



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

**Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa**

TREBALL DE FI DE GRAU

Grau en Enginyeria Electrònica Industrial i Automàtica

Grau en Enginyeria Elèctrica

DISSENY I POSADA EN MARXA D'UN SISTEMA SCADA BASAT EN PLATAFORMES IOT

ANNEX

Autor: Roger Bautista Florensa

Directors: Álvaro Luna Alloza i Néstor Berbel Artal

Convocatòria: Juny 2020

SUMARI

Annex 1. Diagrama de Gantt del projecte

Annex 2. Manual d'ús del Power SCADA Operation

Annex 3. Manual d'ús de Microsoft Azure

Annex 4. Informe amb Power BI

Annex 5. Codi de programació del SCADA

Annex 6. Codi de TwinCAT 3 per llegir del canal Modbus

Annex 7. Codi de TwinCAT 3 per enviar dades a IoT Hub utilitzant MQTT

Annex 8. Manual de connexió i enviament de dades a IoT Hub de Beckhoff

Annex 9. Taula de senyals del SCADA

ANNEX 1

DIAGRAMA DE GANTT DEL PROYECTO

En aquest document es planteja el diagrama de Gantt per a la realització del projecte, explicant les diferents tasques realitzades i la dependència entre elles. D'aquesta manera es comprèn la dimensió i durada del projecte, exposant també labors que no s'han pogut executar en el projecte per manca recursos, però que en el desenvolupament complet d'aquest serien necessàries.

D'altra banda, el plantejament que aquí es proposa va enfocat únicament a les competències del projecte d'estudi, doncs no s'incidirà en el desenvolupament del sistema de control mitjançant PLCs, per exemple.

A continuació, en una taula resum, es manifesten les tasques del projecte i les dependències entre elles.

| TASCA | DESCRIPCIÓ | DEPENDÈNCIES |
|-------|---|------------------------|
| 1 | Estudi i investigació de l'estat de l'art dels sistemes SCADA aplicats al sector energètic. | 2 |
| 2 | Comprensió del projecte, la seva dimensió i les tecnologies que s'hi apliquen. | 1; 4; 6; 7; 8 |
| 3 | Familiarització amb Power SCADA Operation. | - |
| 4 | Desenvolupament del SCADA amb PSO. | 1; 2; 3 |
| 5 | Test de funcionament del SCADA amb PSO. | 2; 4 |
| 6 | Curs de TwinCAT 3 de Beckhoff. | 2 |
| 7 | Investigació sobre plataformes en el núvol i serveis que utilitzen. Solucions possibles sobre control i monitoratge | 2; 8 |
| 8 | Investigació sobre comunicacions entre PLC/SCADA i plataformes en el núvol. | 2; 7 |
| 9 | Desenvolupament de solució per comunicar-se amb Azure. | 2; 6; 7; 8 |
| 10 | Emmagatzematge de dades en el núvol. | 7; 8; 9 |
| 11 | Investigació sobre opcions de visualització localment i en el núvol de les dades emmagatzemades en el núvol. | 7; 8; 9; 10 |
| 12 | Generació de pantalles de monitoratge amb Grafana. | 2; 10; 11; |
| 13 | Investigació de possibilitats de generació d'informes utilitzant Power BI. | 4; 7; 8; 9; 10 |
| 14 | Generació d'un de prova HMI amb TwinCAT HMI. | 2; 4; 6 |
| 15 | Muntatge i posta en marxa del projecte en un entorn físic real. | 2; 4; 5; 9; 10; 12; 13 |
| 16 | Redacció de la memòria i el pressupost | Totes les tasques |

[illegible]

ANNEX 2

MANUAL D'ÚS DEL POWER SCADA OPERATION

Índex

| | |
|--|----|
| Índex de figures | 2 |
| 1. Objecte del document..... | 4 |
| 2. Configuració bàsica del Power SCADA Operation | 4 |
| 3. Power SCADA Studio | 5 |
| 3.1 Projects..... | 6 |
| 3.2 Topology..... | 12 |
| 3.3 System Model..... | 13 |
| 3.4 Visualization | 17 |
| 3.5 Security..... | 18 |
| 3.6 Compilar i córrer el projecte actual..... | 18 |
| 3.7 Citect Graphic Builder | 19 |
| 3.8 Citect Cicode Editor | 19 |
| 4. Profile Editor | 20 |
| 4.1 Configuració | 20 |
| 4.2 Crear un projecte en el PSO mitjançant Profile Editor..... | 25 |
| 5. Generació d'elements i pantalles..... | 26 |
| 5.1 Pantalles i models..... | 26 |
| 5.2 Elements..... | 27 |
| 5.3 Genie i Super Genie..... | 33 |
| 6. Programació Cicode | 37 |
| 7. Power SCADA Operation Help..... | 38 |

Índex de figures

| | |
|---|----|
| Il·lustració 1: Aplicacions instal·lades amb el PSO | 4 |
| Il·lustració 2: Pantalla principal en el Power SCADA Studio | 5 |
| Il·lustració 3: Menú desplegable en el Power SCADA Studio | 6 |
| Il·lustració 4: Desplegable en pestanya Add | 7 |
| Il·lustració 5: Primera pantalla a configurar en el Project Setup | 7 |
| Il·lustració 6: Configuració del servidor a connectar-se en el projecte..... | 8 |
| Il·lustració 7: Configuració dels usuaris i els rols d'aquests en el Project Setup | 8 |
| Il·lustració 8: Configuració de les pantalles..... | 9 |
| Il·lustració 9: Associació de les variables i dispositius en el projecte..... | 10 |
| Il·lustració 10: Finestra emergent per crear un projecte en blanc | 10 |
| Il·lustració 11: Desplegable en la pestanya Remove | 11 |
| Il·lustració 12: Opcions en la pestanya Backup | 11 |
| Il·lustració 13: Dispositius configurats en un projecte | 12 |
| Il·lustració 14: Variables provinents d'un dispositiu (PLC) | 14 |
| Il·lustració 15: Variables locals | 14 |
| Il·lustració 16: Configuració del menú de la interfície HMI..... | 17 |
| Il·lustració 17: Configuració dels diferents rols i privilegis..... | 18 |
| Il·lustració 18: Córrer el projecte | 19 |
| Il·lustració 19: Compilar | 19 |
| Il·lustració 20: Accedir al Citect Graphic Builder | 19 |
| Il·lustració 21: Profile Editor, pestanya de Device Type Tags | 20 |
| Il·lustració 22: Edició dels tags/variables, dividint en tres parts la finestra per separar en parts l'explicació | 21 |
| Il·lustració 23: Finestra per crear o editar tags/variables | 22 |
| Il·lustració 24: Configuració de les alarmes d'una senyal | 23 |
| Il·lustració 25: Pestanya on es configuren els dispositius | 24 |
| Il·lustració 26: Selecció d'un model (Template) amb la fi de dissenyar una nova pantalla o modificar-lo | 26 |
| Il·lustració 27: Pantalla amb model donat per defecte amb el PSO | 27 |
| Il·lustració 28: Editor de propietats d'un rectangle | 28 |
| Il·lustració 29: Pestanya que permet condicionar quan no ha de ser visible el rectangle | 29 |
| Il·lustració 30: Configurar si es vol moure l'objecte en funció del valor d'una expressió | 29 |

| | |
|--|----|
| Il·lustració 31: Configuració per determinar l'ample d'un objecte en funció del valor d'una expressió | 30 |
| Il·lustració 32: Configurar el color de l'interior d'un objecte en base una certa condició | 31 |
| Il·lustració 33: Propietat Input de la configuració..... | 32 |
| Il·lustració 34: Configuració perquè una variable passi a tenir un nou valor quan es compleix una certa comanda..... | 32 |
| Il·lustració 35: Configurar quins usuaris tenen accés a l'objecte | 33 |
| Il·lustració 36: Configurar el Genie perquè es posi en verd quan la variable que s'associarà a ell valgui 10 | 34 |
| Il·lustració 37: Configuració del Super Genie quan es prem el botó 'Start' | 35 |
| Il·lustració 38: Configurar en les propietats del Super Genie l'associació de variables | 35 |
| Il·lustració 39: Relació entre les variables del Genie i les del Super Genie | 36 |
| Il·lustració 40: Finestra que s'obra en la pantalla del SCADA per configurar els valors del Genie | 37 |











1. Objecte del document

La motivació d'aquest document es servir de referència bàsica per l'aprenentatge i ús del Power SCADA Operation, ja que existeix escassa informació o exemples sobre com utilitzar-lo.

D'aquesta manera, amb aquest manual es comprendrà amb més facilitat el programa i com crear un projecte bàsic amb aquest.

2. Configuració bàsica del Power SCADA Operation

Primerament, introduït el programa, quan s'instal·la, es genera una carpeta en l'escriptori amb les següents aplicacions.

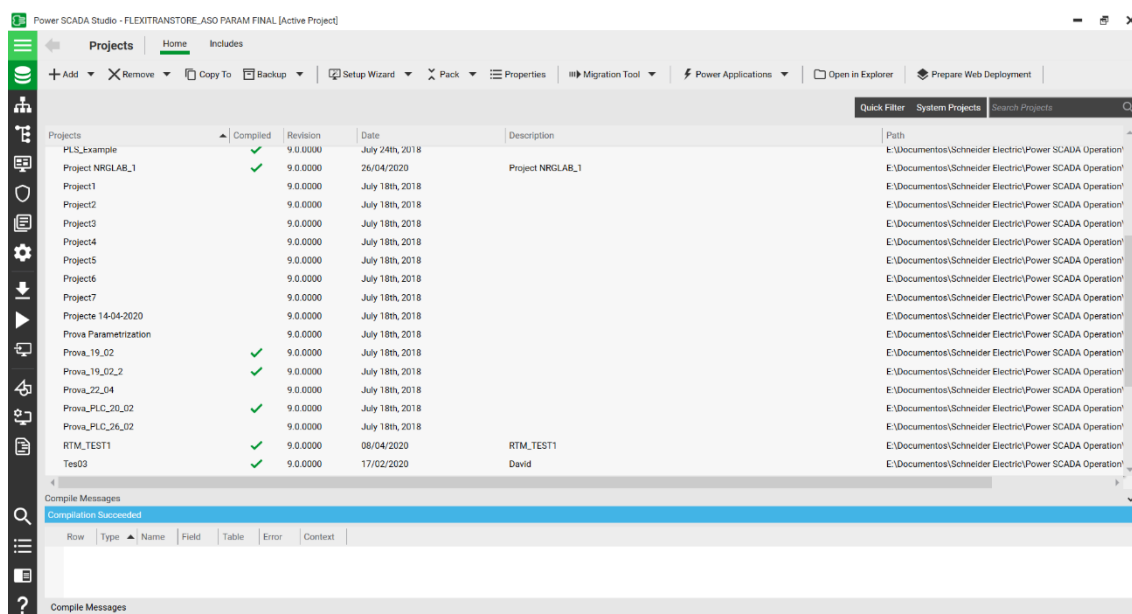
| Nombre | Fecha de modificación | Tipo | Tamaño |
|--|-----------------------|-----------------------|--------|
|  Application Config Utility | 05/02/2020 21:02 | Acceso directo | 2 KB |
|  IO Device Manager | 05/02/2020 21:01 | Acceso directo | 2 KB |
|  LiveView | 05/02/2020 21:02 | Acceso directo a l... | 1 KB |
|  Power SCADA Operation Runtime Manag... | 05/02/2020 21:03 | Acceso directo | 2 KB |
|  Power SCADA Operation Runtime | 05/02/2020 21:03 | Acceso directo | 2 KB |
|  Power SCADA Studio | 05/02/2020 21:03 | Acceso directo | 2 KB |
|  Profile Editor | 05/02/2020 21:01 | Acceso directo | 2 KB |
|  Project Setup | 05/02/2020 21:01 | Acceso directo | 2 KB |
|  Reporting | 05/02/2020 21:02 | Acceso directo a l... | 1 KB |
|  Service Display Client (Control) | 05/02/2020 20:59 | Acceso directo | 2 KB |
|  Service Display Client (View only) | 05/02/2020 20:59 | Acceso directo | 2 KB |
|  Template Editor | 05/02/2020 21:02 | Acceso directo | 2 KB |

Il·lustració 1: Aplicacions instal·lades amb el PSO

Moltes d'aquestes aplicacions no fa falta utilitzar-les, i altres ja van integrades en altres. De totes maneres, no s'incidirà en algunes d'elles perquè no és estrictament necessàries per poder treballar amb el programa.

3. Power SCADA Studio

Seguint en la línia anterior, l'element bàsic i principal en el PSO és l'aplicació que apareix en la il·lustració 1 com "Power SCADA Studio". Accedint-hi, es visualitza la següent interfície.



Il·lustració 2: Pantalla principal en el Power SCADA Studio

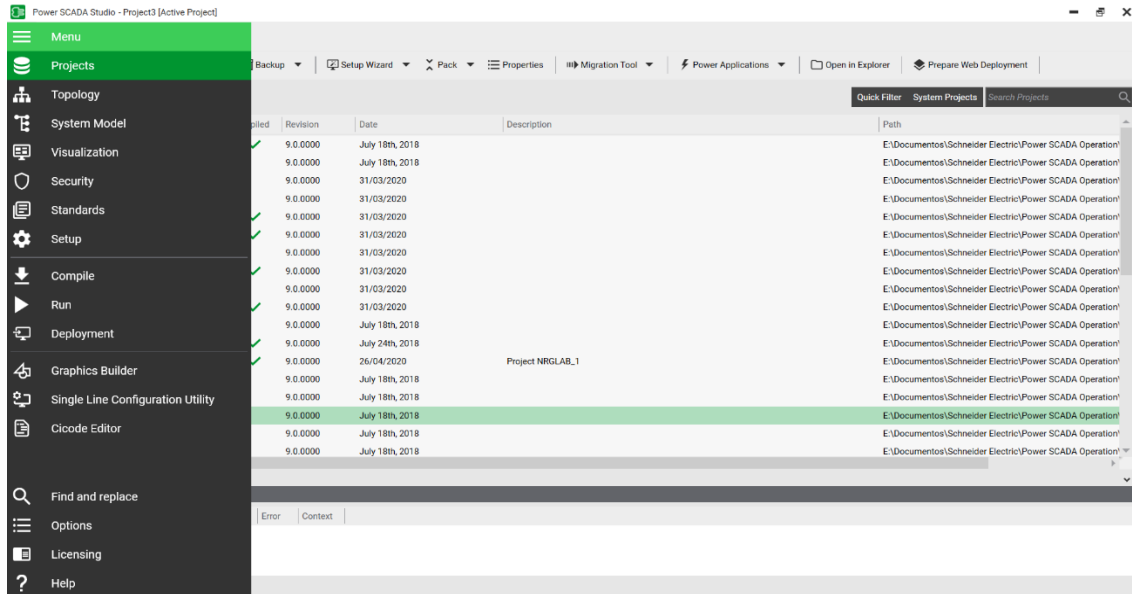
Des d'aquest programa s'accedeix a la resta que seran tractats en aquest bloc.

En la imatge anterior veiem la interfície principal del programa, on en primer pla tenim tots els projectes existents que s'han anat creant en el nostre ordinador.

En la part inferior, veiem un espai que va dedicat als diferents missatges d'error i d'avertència quan compilem un projecte.

En la part superior, s'hi aprecien diferents botons i/o opcions que ja seran analitzats posteriorment.

Finalment, a la part lateral esquerra hi ha el menú que ens permet moure'ns entre les diferents opcions de configuració del programa. Quan es desplega, la seva visualització és la que es veu en la il·lustració 3.



Il·lustració 3: Menú desplegable en el Power SCADA Studio

A continuació, s'aniran exposant les diferents pestanyes del menú i que s'hi configura en cada cas.

3.1 Projects

Aquest apartat del menú és la pantalla inicial que es mostra en la il·lustració 3.

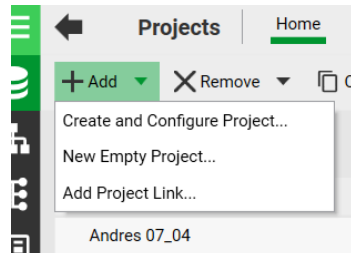
Tal com s'ha comentat anteriorment, en el pla principal veiem els diferents projectes existents on s'hi indica el nom, si està compilat o no, la data de creació, una possible descripció i finalment la seva localització en l'equip

Si el projecte està ombrejat en verd, indica que és el projecte existent el qual es pot editar o fer-lo córrer.

En la part superior, s'observen diferents opcions/botons que seguidament seran explicats amb detall els més importants o comuns.

I. Add

La primera opció és el botó Add que permet afegir nous projectes. Presenta tres opcions de configuració on en cada cas el plantejament és diferent.

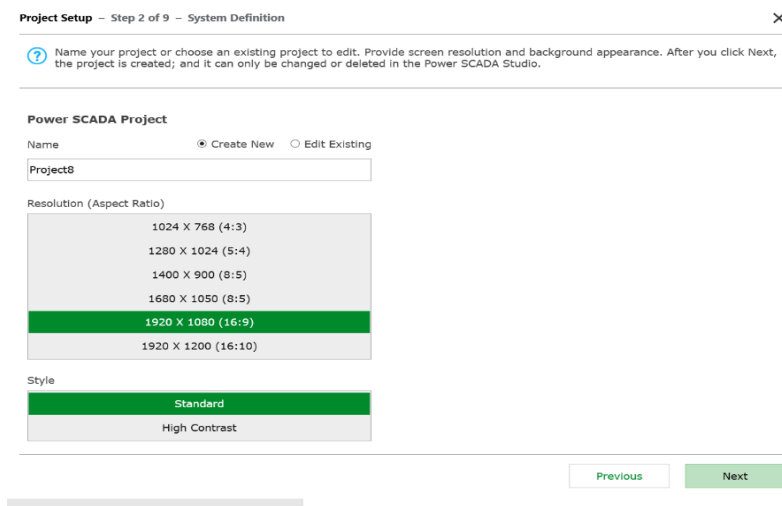


Il·lustració 4: Desplegable en pestanya Add

Create and Configure Project

Aquesta opció obre el Project Setup, una de les aplicacions que apareixien en la carpeta del PSO indicada en la il·lustració 1, s'ha de configurar en diferents passos.


- 1) Inicialment, permet poder elegir entre crear un projecte nou o editar-ne un d'existent, modificant diferents aspectes de la seva configuració in situ.



Il·lustració 5: Primera pantalla a configurar en el Project Setup

- 2) Serveix per indicar la IP del servidor a connectar-se, tenint en compte que l'SCADA actua com a client. També podem afegir diferents IPs si volem que l'SCADA llegeixi de diferents servidors.

Project Setup – Step 3 of 9 – Servers X

 Enter the address information for the servers in this project. If two or more servers are in the Default_Starter.CTZ starter project file, only the address and user account fields for Advanced Reports and Dashboards will be available.

Topology

Server Name or IP Address

Note: Choose a different server name or IP address if you wish to enable redundant system capabilities.

☐ Redundant System

Standby Server Name or IP Address

☐ Advanced Reports and Dashboards

Advanced Reports Server Name or IP Address


User Name Password Confirm Password

[Previous](#) [Next](#)

Il·lustració 6: Configuració del servidor a connectar-se en el projecte

- 3) El següent pas ens permet crear diferents usuaris, amb rols i contrasenyes segons les necessitats que tinguem en l'SCADA. Es crearan usuaris diferents i amb permisos diferents si per exemple es vol que uns usuaris puguin tenir accés a certs elements/pantalles i altres no.

Project Setup – Step 4 of 9 – Users X

 Add the Power SCADA user account information for each user who will access this project. Each user must be assigned to a role. Each role can also be a member of a pre-established Windows group.

Power SCADA Users

[Add User](#)

| User Name | Role | Password | Confirm Password | Full Name (optional) |
|-----------|-------|----------|------------------|----------------------|
| bol | Role0 | •••••••• | •••••••• | aol |

[Delete Selected](#)

Windows Authentication - Active Directory (optional)


| Role | Windows Group |
|------------|---------------|
| Controller | |
| Operator | |
| Role0 | |

[Previous](#) [Next](#)

Il·lustració 7: Configuració dels usuaris i els rols d'aquests en el Project Setup

- 4) En aquest pas, es pot elegir quins tipus de menús es volen tenir i quina pantalla es vol obrir d'inici. També es contempla l'opció de tenir més d'un monitor i per tant, quines pantalles s'obriran en primer moment en cada cas.

Project Setup – Step 5 of 9 – Display: Menus and Display Pages ✕

 Determine the menu items that will display on the top-level tabs of the runtime screen. Also, you can choose the landing page that will initially display on each monitor in a multiple-monitor system.

HMI Menus

- ☐ Home
- ☐ Graphics
- ☐ One-line
- ☐ Alarms / Events
- ☐ Analysis (Process Analyst, Instant Trend, Waveform, Tag Viewer)
- ☐ Advanced Reporting
- ☐ Dashboards

Monitors

Total Monitors – +

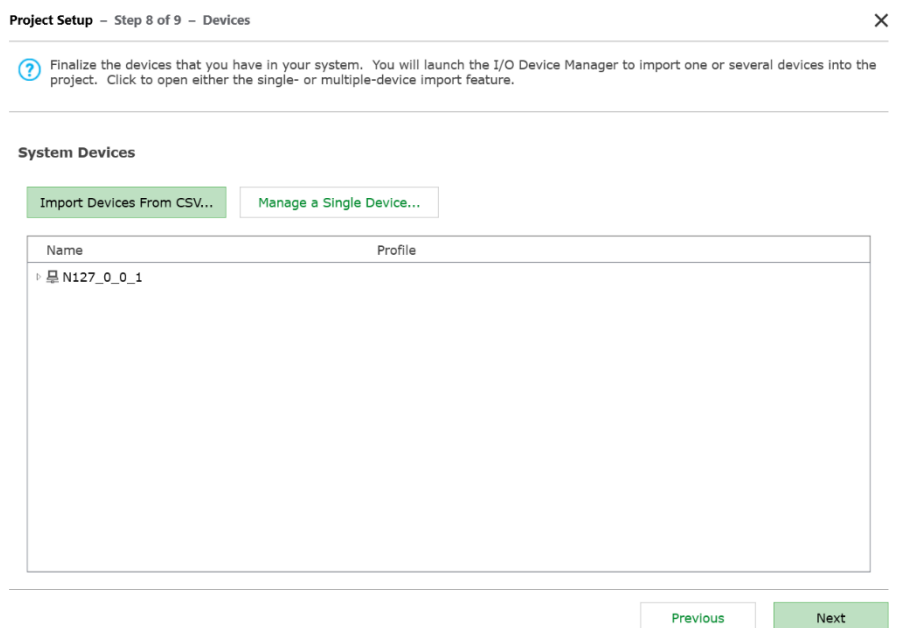
Runtime Landing Page

| | |
|---|---|
| Monitor 1 | Monitor 2 |
| <input type="text" value="PLSStartup"/> | <input type="text" value="PLSStartup"/> |
| Monitor 3 | Monitor 4 |
| <input type="text" value="PLSStartup"/> | <input type="text" value="PLSStartup"/> |
| Monitor 5 | Monitor 6 |
| <input type="text" value="PLSStartup"/> | <input type="text" value="PLSStartup"/> |
| Monitor 7 | Monitor 8 |
| <input type="text" value="PLSStartup"/> | <input type="text" value="PLSStartup"/> |

Previous Next

Il·lustració 8: Configuració de les pantalles

- 5) Seguidament apareix una pantalla resum amb tota la configuració que s'ha realitzat fins ara.
- 6) Per acabar, apareix una pantalla corresponent al 'Project Setup' que ens permet seleccionar dispositius amb senyals associades. Aquest procés s'explica detalladament en el capítol dedicat al Profile Editor.

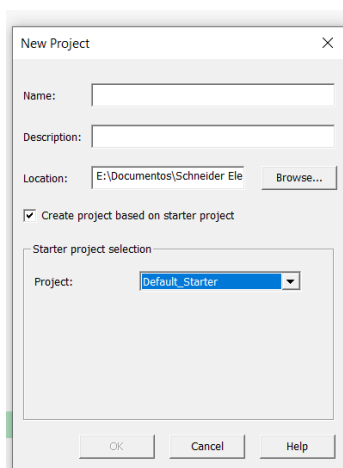


Il·lustració 9: Associació de les variables i dispositius en el projecte

New Empty Project

Permet crear un projecte en blanc, sense cap configuració com en el cas de la prèvia. Aquesta però, es pot fer més endavant seleccionant l'opció d'editar el projecte.

En aquest cas, únicament indiquem el nom del projecte, una possible descripció i la seva localització en l'equip.



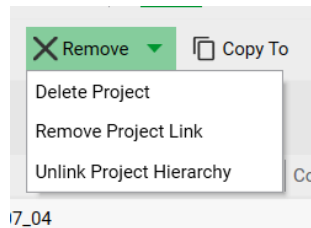
Il·lustració 10: Finestra emergent per crear un projecte en blanc

Add Project Link

Serveix per importar un projecte que estigui localitzat en una altra carpeta, per el cas de tenir-lo en un USB o en una xarxa remota.

II. Remove

Aquesta opció implica eliminar un o varis projectes. En aquest cas també hi ha diverses opcions.



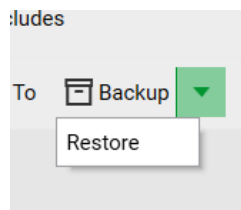
Il·lustració 11: Desplegable en la pestanya Remove

- **Delete Project**: Elimina el projecte seleccionat.
- **Remove Project Link**: Enfocat als projectes que s'han creat mitjançant un link.
- **Unlink Project Hierarchy**: Es basa en eliminar el projecte associat com en el cas anterior però també tots els projectes fill d'aquest primer.

III. Backup

En el cas de prémer a Backup, es generarà un document .ctz en la carpeta desitjada on hi haurà el projecte comprimit.

En el cas de prémer Restore, és per fer el procés invers. A partir d'un arxiu .ctz importem un projecte.



Il·lustració 12: Opcions en la pestanya Backup

3.2 Topology

En aquesta pestanya del menú, s'hi inclouen i configuren tot el que està relacionat amb màquines físiques (ordinadors), *clústers*¹, servidors i dispositius d'E/S.

Trobem diferents apartats a poder configurar en aquest submenú, seguidament es parlarà dels principals.

I. Computers

En aquest cas, visualitzem els ordinadors i/o clústers que tinguem en el projecte.

II. Edit

En aquest apartat es poden afegir i modificar clústers al projecte.

III. I/O Devices

Aquí es configuren els diferents dispositius als que l'SCADA està connectat. Es pot donar el cas que estigui connectat a diferents PLCs, RTUs, sensors o altres elements, i llegeixi senyals de cada un. En aquest cas tindríem un dispositiu per cada un, on s'hi configura el protocol de comunicacions elegir i el port que s'utilitza com a elements cabdals.

| Row | Server Name | Name | Number | Address | Protocol | Port Name | Startup Mode | Priority | Memory | Read-Only | Comment |
|-----|-------------|----------------|--------|---------|-----------|--------------|--------------|----------|--------|-----------|---------|
| 1 | IOLServer1 | zOL | 1 | | IEC61850N | | Primary | 1 | TRUE | | |
| 2 | IOLServer1 | NetworkTagsDev | 2 | | IEC61850N | | Primary | 1 | TRUE | | |
| 3 | IOLServer1 | NRGLAB_Device | 3 | 1 | MODNET1 | PORT2_BOARD1 | Primary | 1 | | | |
| 4 | IOLServer1 | IOPrueba | 4 | 1 | MODNET1 | PORT2_BOARD1 | Primary | 1 | | | |

Il·lustració 13: Dispositius configurats en un projecte

Els principals protocols amb que treballa el PSO són:

- ASCII: per a comunicacions sèries simples.
- Modbus: protocol sèrie molt usat en automatització.

¹ Clúster: Sistema de processament en paral·lel o distribuït on cada element es coneix com a node.

- DNP 3.0: protocol per xarxes distribuïdes.
- BACNet: enfocat per automatització d'edificis a nivell de control.
- OPC: protocol aplicar per transmetre dades a nivell PC.
- IEC8010-5: implica un perfil de comunicacions per enviar missatges bàsics de telecontrol.
- EIB: bus europeu.
- Profibus: protocol per a bus de camp en automatització.
- SNMP: protocol aplicat en xarxes de dispositius.

Tenim opció de poder importar dispositius des d'un document CSV o bé des d'un Profile Editor, que això és un procés que s'explicarà més endavant.

IV. Components and Mapping

Troblem aquí la possibilitat de veure les IPs i els ports que s'utilitzen en cada cas. Tenim un menú desplegable en el lateral esquerra per poder moure's entre diferents opcions.

3.3 System Model

En aquesta opció del menú tenim la possibilitat de configurar com a elements destacats l'equipament, les variables, alarmes, variables de tendència i els acumuladors.

I. Equipment

En Citect SCADA, i per inri, en el PSO, s'utilitza el terme d'equipaments per referir-se a una arquitectura basada en objectes orientats per a un projecte que es pot referir a maquinària o processos.

Els equipaments creen agrupacions lògiques que permeten agrupar el projecte d'una forma jerarquitzada.

Per tant es poden crear equipaments en funció dels elements, agrupant així les diferents senyals, alarmes, permisos i comunicacions que hi té associats.

II. Variables

Es pot definir com a variable un element anomenat i localitzat en la memòria del ordinador on s'hi poden emmagatzemar dades d'un tipus definit.

El PSO permet definir variables associades a dispositius, i per altra banda, locals, que únicament fan referència al SCADA i no van associades amb ningun dispositiu extern, però dins el programa es poden fer servir igual que una variable qualsevol.

| Row | Item Name | Tag Name | Cluster | I/O Device | Data Type |
|-----|-----------|--|---------|---------------|-----------|
| 1 | ce | NRGLAB_Device/FLCmdMB_COMM | c1 | NRGLAB_Device | INT |
| 2 | ce | NRGLAB_Device/FLCmdMB_IE | c1 | NRGLAB_Device | INT |
| 3 | ce | NRGLAB_Device/FLCmdMB_FR | c1 | NRGLAB_Device | INT |
| 4 | ce | NRGLAB_Device/FLCmdMB_MFA | c1 | NRGLAB_Device | INT |
| 5 | ce | NRGLAB_Device/FLCmdMB_VR | c1 | NRGLAB_Device | INT |
| 6 | ce | NRGLAB_Device/FLCmdMB_AddServ | c1 | NRGLAB_Device | INT |
| 7 | ce | NRGLAB_Device/FLCmdEAP_COMM | c1 | NRGLAB_Device | INT |
| 8 | ce | NRGLAB_Device/FLCmdEAP_IE | c1 | NRGLAB_Device | INT |
| 9 | ce | NRGLAB_Device/FLCmdEAP_FR | c1 | NRGLAB_Device | INT |
| 10 | ce | NRGLAB_Device/FLCmdEAP_MFA | c1 | NRGLAB_Device | INT |
| 11 | ce | NRGLAB_Device/FLCmdEAP_VR | c1 | NRGLAB_Device | INT |
| 12 | ce | NRGLAB_Device/FLCmdEAP_AddServ | c1 | NRGLAB_Device | INT |
| 13 | ce | NRGLAB_Device/SLCmd_COMM | c1 | NRGLAB_Device | INT |
| 14 | ce | NRGLAB_Device/SLCmd_RVM | c1 | NRGLAB_Device | INT |
| 15 | ce | NRGLAB_Device/SLCmd_PSM | c1 | NRGLAB_Device | INT |
| 16 | ce | NRGLAB_Device/SLCmd_RRE | c1 | NRGLAB_Device | INT |
| 17 | ce | NRGLAB_Device/SLCmd_PCM | c1 | NRGLAB_Device | INT |
| 18 | ce | NRGLAB_Device/SLCmd_PBP | c1 | NRGLAB_Device | INT |
| 19 | ce | NRGLAB_Device/SLCmd_AddServ | c1 | NRGLAB_Device | INT |
| 20 | ce | NRGLAB_Device/FLCparamMB_COMM_NominalPower | c1 | NRGLAB_Device | REAL |
| 21 | ce | NRGLAB_Device/FLCparamMB_COMM_NominalVoltage | c1 | NRGLAB_Device | REAL |

II-lustració 14: Variables provinents d'un dispositiu (PLC)

| Row | Name | Data Type | Array Size | Eng Units | Format | Comment | Zero Scale | Full Scale |
|-----|-----------------|-----------|------------|-----------|----------|---------|------------|------------|
| 1 | aComand | INT | 20 | | #### | | | |
| 2 | aParam | REAL | 324 | | #####.## | | | |
| 3 | aMin | REAL | 324 | | #####.## | | | |
| 4 | aMax | REAL | 324 | | #####.## | | | |
| 5 | aDef | REAL | 324 | | #####.## | | | |
| 6 | aMin_initial | REAL | 324 | | #####.## | | | |
| 7 | aMax_initial | REAL | 324 | | #####.## | | | |
| 8 | aDef_initial | REAL | 324 | | #####.## | | | |
| 9 | aOp_IE | DIGITAL | 24 | | ## | | | |
| 10 | aOp_FR | DIGITAL | 24 | | ## | | | |
| 11 | aOp_MF | DIGITAL | 24 | | ## | | | |
| 12 | aOp_VR | DIGITAL | 24 | | ## | | | |
| 13 | aOp_RVM | DIGITAL | 24 | | ## | | | |
| 14 | aOp_PSM | DIGITAL | 24 | | ## | | | |
| 15 | aOp_RRE | DIGITAL | 24 | | ## | | | |
| 16 | aOp_PCM | DIGITAL | 24 | | ## | | | |
| 17 | aOp_PBP | DIGITAL | 24 | | ## | | | |
| 18 | Op_Pref | REAL | 24 | kW | #####.## | | | |
| 19 | Op_Qref | REAL | 24 | kVar | #####.## | | | |
| 20 | FLC_Comms | DIGITAL | | | ## | | | |
| 21 | Op_ProgMan_Mode | DIGITAL | | | ## | | | |

II-lustració 15: Variables locals

A continuació es fa una explicació dels diferents camps a omplir en cada cas.

Variables del dispositiu:

Per les variables que es comuniquen amb un dispositiu extern cal definir els següents camps:

- Equipment: Indica l'equip a que pertany la senyal, si únicament existeix un únic dispositiu no és necessari incloure-ho.
- Tag Name: S'assigna un nom a la variable del dispositiu.
- Clúster: S'indica el clúster a què pertany.
- I/O Device: S'hi ha associat el dispositiu corresponent.
- Data Type: Tipus de dades de la senyal, s pot elegir entre: BCD, Byte, Digital, Int, Long, LongBCD, Real, String, Uint i Ulong.
- Adres: Direcció des d'on el llegeix i s'escriu en la senyal. Depèn del canal de comunicacions que s'utilitzi.

La resta de camp no són obligatoris d'omplir.

Variable local:

Quan es crea una variable local cal definir els següents camps:

- Name: Nom que s'assigna a la variable local.
- Data type: Igual que en les variables del dispositiu.
- Array: Si la variable local és un vector o matriu, cal definir quines dimensions té i no pot superar el total dels 32766 elements. La matriu pot ser de fins a dos dimensions, si es dona el cas cal definir-ho separant files i columnes amb una coma, per exemple: "20,30". Si la variable és d'un únic element no cal posar-hi res.
- Format: Si es vol definir un format de dades tenint en compte decimals per exemple.

III. Alarms

Una alarma indica una condició anormal que ha passat en el SCADA.

Una alarma és un element principal és un sistema de control, i per tant, s'han de configurar de tal manera que es reportaran a l'operador perquè respongui adequadament.

En el cas del PSO hi ha diferents tipus de variables que es poden definir i utilitzar. A continuació es defineixen les més habituals únicament.

Alarmes Digitals

Té dos estats (ON i OFF). Canvia d'estat en funció de fins a dos variables també digitals.

Es pot especificar un cert retard per una alarma digital per tal d'assegurar que la condició que activa l'alarma es perpetua.

Alarmes Analògiques

Aquest tipus s'activa quan una variable analògica supera uns certs valors llindar.

Tenen certes combinacions.

- High i High-High: S'activa quan la variable supera un dels límits superiors marcats. High-High indica major incidència que High, i el temps de retard configurable, ha de ser més immediat en el primer cas que el segon.
- Low i Low-Low: S'activa quan la variable rabassa uns límits inferior marcats. Segueix el mateix principi que el cas anterior.
- Desviació: Quan un valor deixa de tenir un valor preestablert.
- Rati de canvi: S'activa si succeeix un canvi brusc d'un valor en un temps especificat.

IV. Trend

Amb les tendències és més fàcil entendre el funcionament i comportament de la planta. Permeten poder visualitzar el valor d'una variable que hi associem en una gràfica.

Així, en aquest cas, ens permet especificar quines dades volem recollir del dispositiu, associant doncs la variable corresponentment. La informació pot ser recollida contínuament o bé, sota esdeveniment, és a dir, quan succeeix quelcom per defecte.

Les dades que es recullen s'emmagatzemen a l'equip, per tant, són consultables més endavant.

Els camps més importants són associar el nom de la variable, el tipus de variable de tendència que es vol i el temps de mostreig de la senyal (cada quan es pren el seu valor).

V. Accumulators

Els acumuladors permeten tenir un control del temps que un cert element i/o senyal ha estat activat o quants cops s'ha activat per exemple.

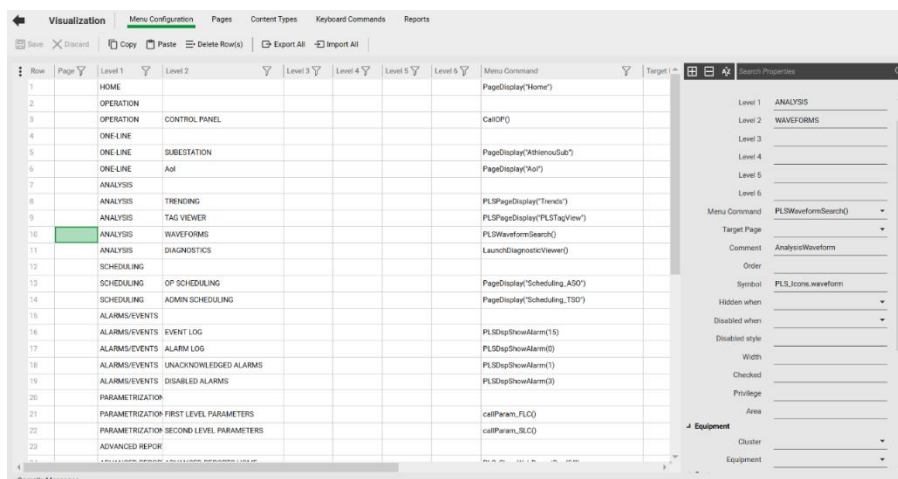
Aquests valors després es poden visualitzar en les pantalles de l'SCADA ja que es guarden com a variables en dispositius d'E/S.

3.4 Visualization

En aquest submenú podem configurar i visualitzar la distribució de les pantalles de l'SCADA

I. Menu Configuration

Aquesta pestanya permet definir una jerarquització del menú en les pantalles per tal de facilitar la navegació a l'operador que controla l'SCADA.



Il·lustració 16: Configuració del menú de la interfície HMI

Utilitzant plantilles en l'SCADA que venen ja incloses amb el programa no fa falta res més que configurar adequadament aquest submenú.

Per tant, s'ha d'establir una jerarquia, sent el primer nivell (Level 1) els elements principals en el menú que es visualitzarà en l'SCADA per poder moure's per les pantalles. Es pot entendre que aquí els títols seran més genèrics. En el segon nivell (Level 2) establim el que vindrien a ser els submenús de cada cas, per tant, determinem una diferenciació més clara. Això es pot anar fent successivament si volem seguir amb la jerarquia fins al nivell 6.

A més a més, s'assignarà una comanda o funció que habiliti accedir a la pantalla desitjada i altres funcions si s'escau, això es fa en el *Menu Command*.

Finalment, tal com veiem en el lateral de la il·lustració 16, es pot associar una icona del Citect Graphic Builder, el qual ja es comentarà en futurs capítols on es troben.

II. Pages

En aquest cas, es presenta un recull de les diferents pantalles que tenim creades en el projecte, els elements d'aquesta pàgina no es poden modificar aquí, si no que s'ha d'accedir a l'editor de pantalles del PSO.

3.5 Security

Configuració dels diferents usuaris i nivells de privilegis/permisos segons cada un.

I. Rols

Tal com s'ha vist en el capítol 3.1.1, quan es configurava el projecte es podien definir diferents usuaris amb certs permisos, aquí s'hi fa el mateix a grans trets.

En aquesta pantalla es poden assignar diferents rols i privilegis.

Els privilegis segueixen una cadena jeràrquica, on el rang més alt és el 8, i l'inferior el 0. Per determinar que un rol té privilegis sobre ell mateix i els seus inferiors cal denotar-ho com per exemple, '8..0', indicant així que el rol en qüestió les seves accions tenen incidència sobre aquells elements que van des del privilegi 8 fins el 0.

Posant un exemple, un rol amb privilegi '3..0', sols podrà actuar sobre elements amb privilegi 3, 2, 1 i 0, però mai superior, els quals no seran accessibles o modificables per ell.

| Row | Role Name | Windows Group | Privileges | Allow RPC | Allow Exec | Manage Users | Comment | Entry Command | Project |
|-----|------------|---------------|------------|-----------|------------|--------------|---------|---------------|--------------|
| 1 | Supervisor | | 1..8 | TRUE | FALSE | TRUE | | | FLEXITRANSTO |
| 2 | Operator | | 1..4 | TRUE | FALSE | FALSE | | | FLEXITRANSTO |
| 3 | Controller | | 1..6 | TRUE | FALSE | FALSE | | | FLEXITRANSTO |
| 4 | User | | 1..2 | TRUE | FALSE | FALSE | | | FLEXITRANSTO |
| 5 | Role0 | | 0 | TRUE | FALSE | FALSE | | | FLEXITRANSTO |

Il·lustració 17: Configuració dels diferents rols i privilegis

II. Users

Seguint amb la línia anterior, en aquest cas es defineixen els usuaris, assignant així un nom d'usuari i contrasenya a cada un que es crea. Juntament, cada usuari té associat un dels rols que es configuren en el cas anterior.

3.6 Compilar i córrer el projecte actual

En el menú lateral que apareix, tal com s'ha explicat en el principi d'aquest capítol, podem veure els dos signes següents:

*Il·lustració 19: Compilar**Il·lustració 18: Córrer el projecte*

El primer permet compilar el projecte seleccionat (en el llistat de projectes, el que està ombrejat en verd). Si no hi ha errors, apareixerà a la part inferior el missatge: “Compilation Succeeded”; si no és així sortirà el llistat d’errors i advertències. Si únicament hi ha advertències, el programa és igualment executable.

El segon signe ens permet córrer o executar el programa, el qual sols és possible si no hi ha errors de compilació.

Prement el botó per córrer el projecte fa que prèviament s’executi la compilació d’aquest.

3.7 Citect Graphic Builder

En el menú lateral trobem un element que en permet accedir al Citect Graphic Builder, el programa que permet dissenyar les diferents pantalles de l’SCADA creant així la interfície HMI.

*Il·lustració 20: Accedir al Citect Graphic Builder*

El programa s’obra en el mateix moment en que s’obra al Power SCADA Studio, i es manté obert mentre aquest ho segon ho està.

En el capítol X s’explicarà en profunditat com funciona.

3.8 Citect Cicode Editor

També des del menú lateral podem accedir al editor de codi del PSO. En ell trobem funcions que ja venen configurades per defecte en el PSO i també hi ha possibilitat d’elaborar el propi codi necessari per l’SCADA que s’està dissenyant.

En el capítol X es comentarà amb més profunditat el seu funcionament.

4. Profile Editor

El Profile Editor és un programa que permet crear dispositius i senyals i integrar els conjunts en projectes.

La lògica que segueix és assimilable a la programació orientada a objectes, assumint que els dispositius són els objectes i cada una de les senyals els seus atributs.

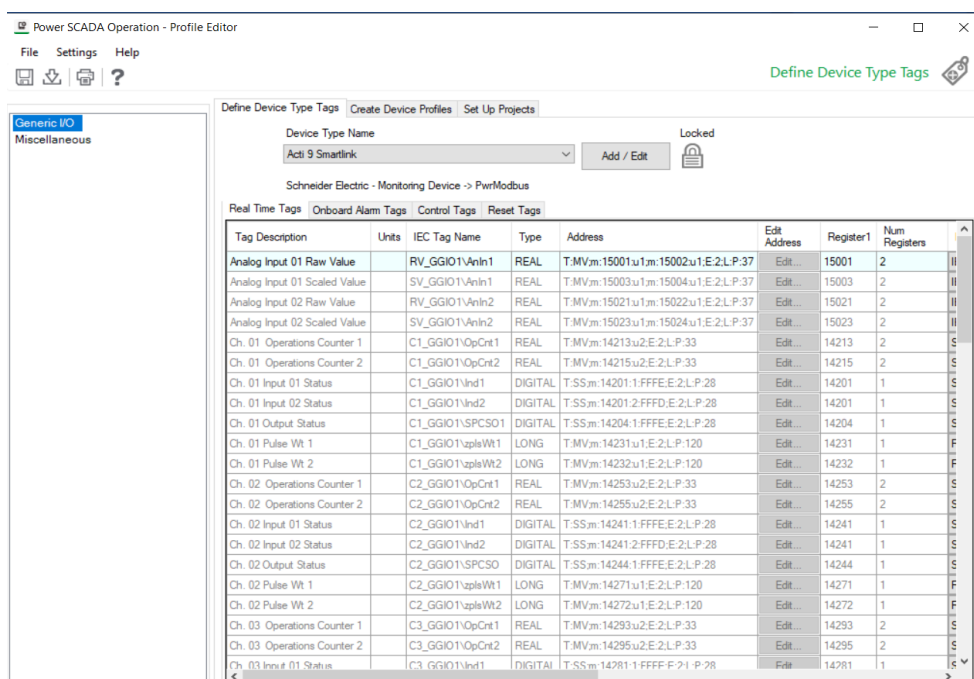
De tal manera, es pot interpretar que podem crear diversos dispositius on contenen les mateixes senyals, únicament que aquestes s'hauran de diferenciar d'alguna, en la direcció del canal de comunicacions per exemple.

En el capítol actual, s'explicarà com es configura adequadament i com utilitzar-lo per tal d'exportar els dispositius creats en un projecte i com es veuen integrats en aquest.

Finalment, i partint de la informació anterior, també s'indicarà com es crea un projecte i s'integren en ell els dispositius del Profile Editor.

4.1 Configuració

Quan s'accedeix al Profile Editor, veiem que es divideix en tres pestanyes: Define Device Type Tags, Create Device Profiles i Set up Projects. Es pot entendre que el primer cas és per configurar o afegir les diferents senyals o tags; el segon per crear els dispositius; per últim, per englobar-ho en un projecte.



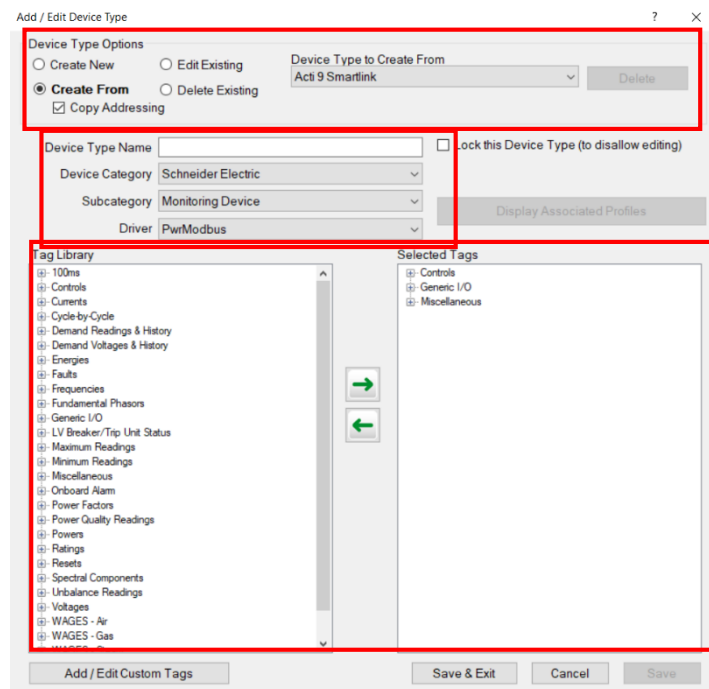
Il·lustració 21: Profile Editor, pestanya de Device Type Tags

I. Definició dels Tags

En el Profile Editor tenim la possibilitat de seleccionar senyals ja creades o bé, crear-ne de noves, adequant sempre les característiques d'aquestes segons el canal de comunicacions que s'utilitzarà en el projecte, el tipus o les unitats, per exemple.

En la il·lustració 21 veiem que es pot configurar la senyal segons si són variables de temps real, alarmes, senyals de control o de reinici. Per tant, s'entén que el programa permet molta versatilitat de configuracions.

Partint del concepte que volem realitzar un nou projecte, s'ha de prémer el botó 'Add/Edit', el qual ens permetrà afegir i eliminar totes aquelles senyals que puguem necessitar.



Il·lustració 22: Edició dels tags/variables, dividint en tres parts la finestra per separar en parts l'explicació

En la part superior es pot apreciar que es poden seleccionar quatre opcions diferents, si es vol crear un nou grup de variables, si es vol editar un existent, si se'n vol crear un partint des d'un altre o si simplement, es vol eliminar-ne un.

Suposant que el volem crear, el següent pas és configurar el segon requadre, on anomenem aquest nou grup de senyals.

En la subcategoria podem elegir en el desplegable entre: Dispositiu per monitorar, PLC i Dispositiu de Protecció. Per defecte, hauríem d'elegir la primera opció, ja que el que volem és poder visualitzar la controlar la senyal en l'SCADA.

Finalment, s'ha de seleccionar un Driver de comunicacions, s'ha de tenir en compte els protocols que s'utilitzen en el projecte per seleccionar-ne un adequadament.

El tercer bloc, inclou totes les senyals. S'agrupen per agrupacions tals com Voltatges, Energia, Potència, en conseqüència implica que estan agrupades segons grups funcionals.

Il·lustració 23: Finestra per crear o editar tags/variables

Per afegir o treure senyals en el nostre projecte hem de buscar en el grup corresponent la que volem, seleccionar-la i prémer la fletxa per passar-la al requadre de la dreta. Si pel cas contrari, volem eliminar alguna senyal del nostre grup o dispositiu, hem de seleccionar-la en el quadre de la dreta i prémer la fletxa central.

Si les senyals que hi ha no ens satisfan o bé, volem que tinguin un nom concret d'acord amb la fulla de senyals del projecte, s'ha d'accedir a un editor mitjançant el botó 'Add/Edit Custom Tag' en la part inferior esquerra.

De nou, en la part superior es planteja la possibilitat, en aquest cas sobre una única variable, de crear-ne una de nova, editar-la, crear-la en base una existent o eliminar-la.

Seguidament, si se'n crea una, s'ha d'introduir un nom i un nom a visualitzar, l'habitual és que els dos siguin el mateix.

Tot seguit, cal indicar a quin grup pertany la senyal. S'ha de tenir en compte la tipologia que és si es vol agrupar adequadament.

Seguidament, es selecciona el tipus, podent elegir entre: Digital, Int, Long, Real o String.

A continuació, si s'escau, s'indica la unitat de la senyal i el format d'aquesta, tenint en compte la magnitud que pot tenir i quina precisió en decimals es vol.

Si es vol, es pot configurar la part esquerra després, on és important que si es vol tenir una alarma sobre la senyal, configurar-la.

☐ Display 'Advanced' filter selections

Real Time Filters Alarm Filters

Categorization N/A

Alarm Type N/A

Alarm Group N/A

Subcategorization N/A

Alarm Level

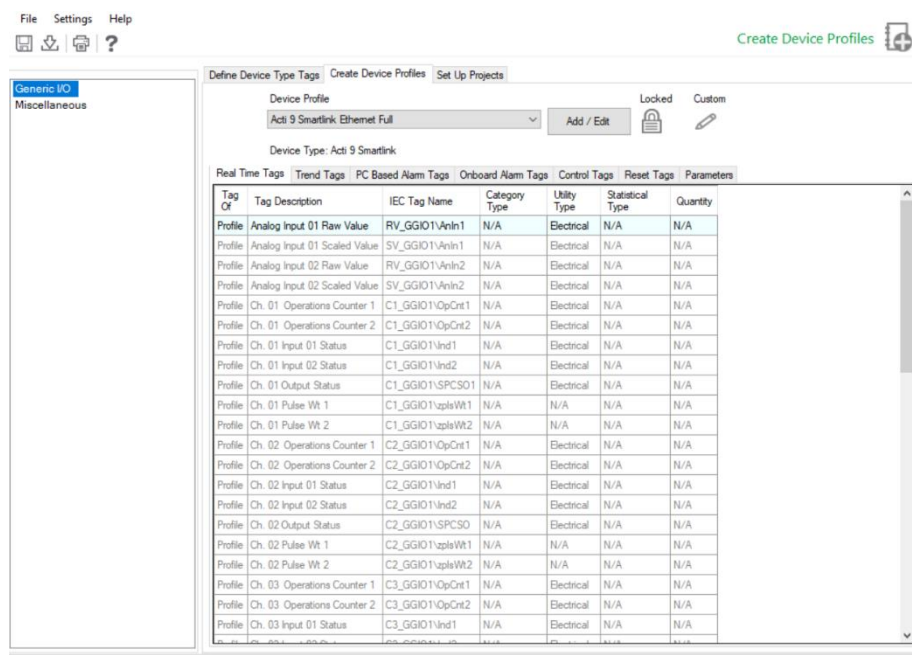
Il·lustració 24: Configuració de les alarmes d'una senyal

Quan s'acaba de configurar, es prem el botó 'Save' si es vol seguir configurant noves senyals o bé, 'Save&Exit' per guardar i tornar a la pantalla anterior. Allà es configura segons s'ha comentat abans.

Després, en la pantalla inicial, cal configurar les senyals afegint-hi la direcció del canal de comunicacions que es llegirà. Cal que el valor sigui el mateix que es posaria el Power SCADA Studio.

II. Definició del Dispositiu

En la següent pestanya, es configura el dispositiu, integrant en ell cada un dels grups de variables que s'hagin pogut crear anteriorment.



Il·lustració 25: Pestanya on es configuren els dispositius

Per afegir o editar un dispositiu hi accedim mitjançant el botó 'Add/Edit', obrint-se així una nova pantalla.

Per crear-ne un de nou cal seleccionar inicialment 'Create New' i prémer a següent. En la nova pantalla que es veu, cal assignar un nom al dispositiu i de nou prémer següent.

A continuació, hem de seleccionar els grups de dispositius que s'han creat o editat en l'apartat anterior i afegir-los i passar a la següent pantalla.

Ara, hem d'afegir de nou tots aquests grups, fent doble clic en cada un d'ells s'afegiran. Així podem veure un recull dels diferents tags que tenim segons tipus de senyal.

Finalment, en la pantalla que apareix, hem de seleccionar grup de senyals per grup, i si volem que les senyals puguin ser visualitzades com a gràfiques en l'Scada cal seleccionar la casella de 'Trend'. Per acabar, cal marcar el quadre de 'Close Wizard' i finalitzar.

Es veurà en la pantalla de la il·lustració 25 totes les senyals que s'inclouen en el dispositiu creat.

III. Associar a un Projecte

En la tercera pestanya, Set Up Projects, cal accedir al 'Add/Edit' i allà seleccionar un projecte creat en PSO i associar-hi els diferents dispositius creats en el cas anterior i guardar.

Seguidament, cal anar a 'File' i en 'Export' fer que les tres opcions estiguin seleccionades.

A continuació, prémer el botó 'Export Project'. S'obrirà un Pop-Up, si hi ha alguna advertència no és problema.

El que s'ha de fer ara és accedir a la icona de carpeta que duu a la localització on s'emmagatzemen els arxius generats en el projecte seleccionat. S'ha de corroborar que en la carpeta 'Profile Editor Export' hi hagi un arxiu .pls, aquest inclou tots els dispositius i senyals que s'han d'integrar en el projecte seleccionat.

4.2 Crear un projecte en el PSO mitjançant Profile Editor

Per poder crear un projecte integrant les diferents senyals i dispositius que es creïn en el Profile Editor cal seguir els següents passos.

- 1) Un cop creats els dispositius en el Profile Editor cal que es tanqui el programa.
- 2) Es crea un nou projecte mitjançant l'opció de 'Create and Configure a Project'.
- 3) Un cop configurat des del Project Setup, s'accedeix al menú Topology i seguidament a la pestanya I/O Devices .
- 4) En el lateral dret s'observa un signe amb tres punts, d'allà sortiran dos opcions, seleccionar la que hi posa 'Device Profile Editor'.
- 5) A continuació s'obra automàticament el Profile Editor. Hem d'anar a la pestanya de 'Set Up Project'.
- 6) Es selecciona l'opció d'editar un projecte existent, on aquí s'ha de buscar que el que s'ha creat recentment.
- 7) S'afegeixen els diferents dispositius necessaris i es treuen aquells que no es vulguin, a continuació es guarda.
- 8) Exportar el projecte.
- 9) De nou, en la pestanya d'I/O Devices de Topology s'accedeix al signe de tres punts entrant a l'opció 'I/O Device Manager'.
- 10) Es selecciona el projecte actual en el desplegable.
- 11) S'accedeix a l'opció 'Manage a Single Device'.
- 12) En la finestra que s'obra com que en aquest cas estem creant el nou projecte s'opta per l'opció 'Create one I/O device'.
- 13) Es selecciona el dispositiu creat en el Device Explorer.
- 14) Es configura segons el requeriments del projecte.
- 15) Quan s'acaba de configurar es veurà que en el projecte del PSO s'han creat els nous dispositius i les variables importats del Profile Editor.

Així, ja es tenen totes les variables i dispositius en projecte correctament configurats. Si cal afegir-ne de nous, creats posteriorment en el Profile Editor, o eliminar-ne s'accedeix al I/O Device Manager i es gestiona d'igual manera.

5. Generació d'elements i pantalles

En aquest capítol s'exposarà com es crea l'element principal de tot SCADA, la interfície principal per controlar i visualitzar els dispositius on està connectat. De tal manera, s'explicarà com es creen els diferents elements i com es configuren.

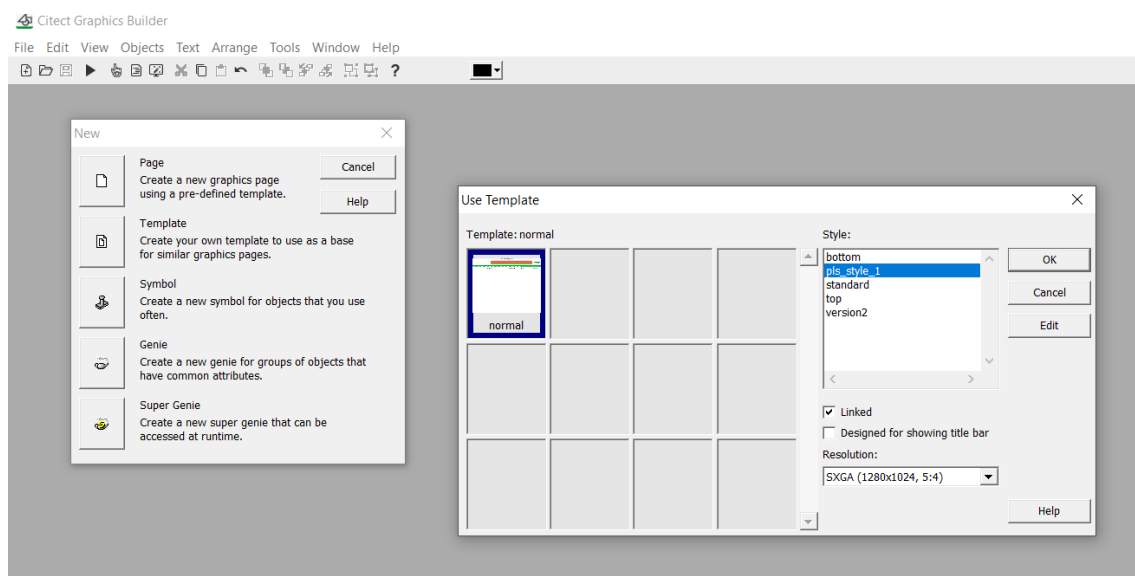
Aquest apartat es centra en la utilització del Citect Graphic Builder.

5.1 Pantalles i models

La visualització que es veu en l'SCADA està gestionada per pantalles per tant, s'han de crear.

Des de 'File' accedint a 'New' s'obra un pop-up on tenim diferents accions, s'elegeix l'opció de 'Page' (pàgina) on a continuació s'obra un nou pop-up on s'elegeix el model. El programa presenta diferents models per defecte amb les diferents resolucions que permet el programa. Alhora, es poden editar aquests models amb l'opció 'Edit'.

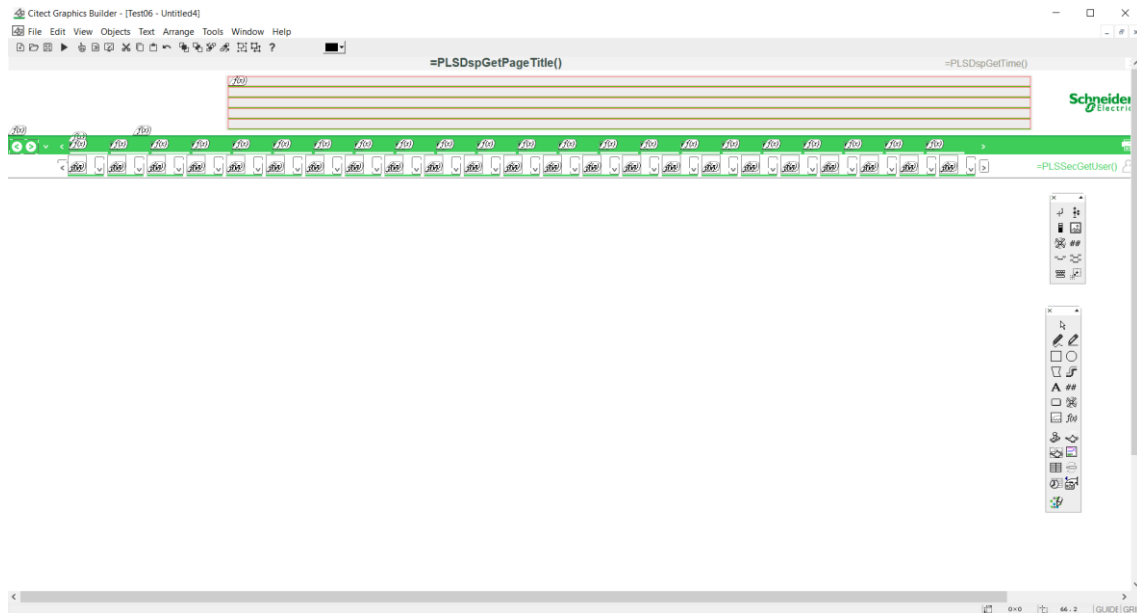
També hi ha l'opció de dissenyar un model des de zero seleccionant l'opció 'Template'.



Il·lustració 26: Selecció d'un model (Template) amb la fi de dissenyar una nova pantalla o modificar-lo

5.2 Elements

La interfície principal (amb un model per defecte) que veiem en l'editor de gràfics del PSO és la següent:

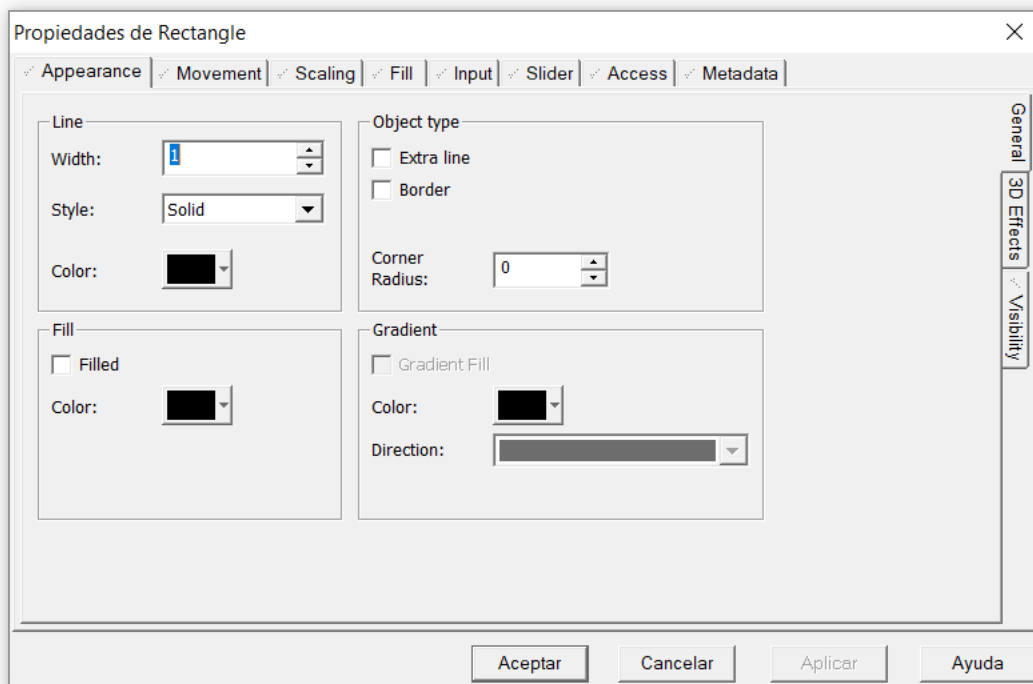


Il·lustració 27: Pantalla amb model donat per defecte amb el PSO

S'observa en el lateral dret dos barres d'eines i d'objectes. Amb aquests és com es dissenya principalment la pantalla. En general, els principals són la línia, el rectangle, el cercle, el botó i el text, tant numèric com cadena de caràcters.

Com a cas exemple, on la majoria d'opcions de configuració són comunes en la resta d'elements, veurem el rectangle.

Quan es dibuixa un en la pantalla s'obra la següent pantalla:

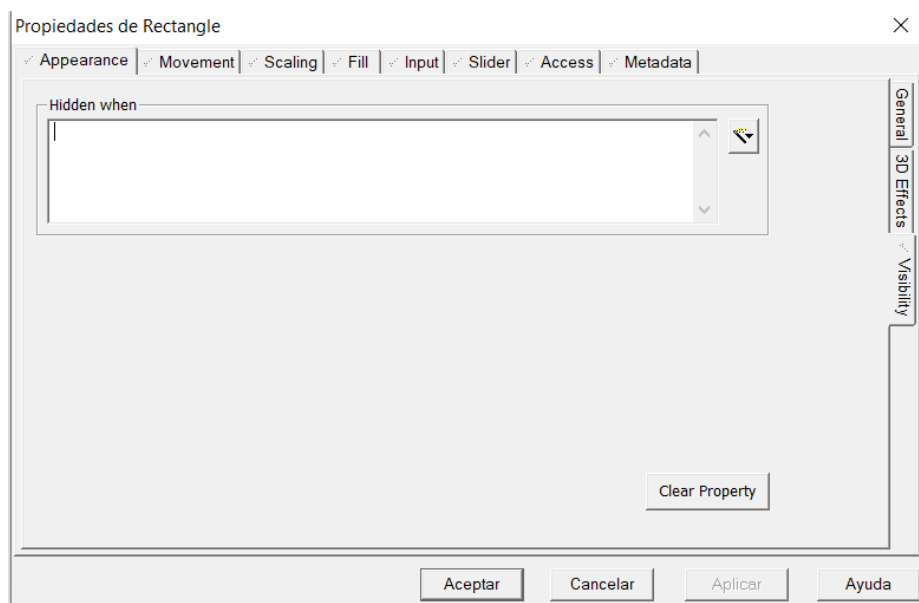


Il·lustració 28: Editor de propietats d'un rectangle

Podent configurar aquí el gruix de les línies, si es vol que el seu interior estigui pintat o si les cantonades es volen arrodonir.

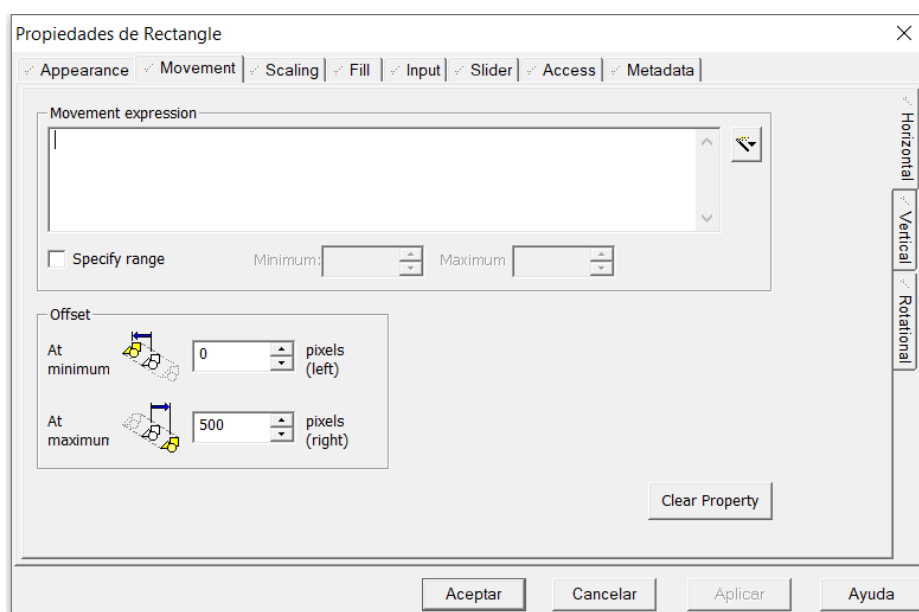
En alguns elements, s'hi pot visualitzar el valor d'una variable que hi associem en 'Display Value'.

En l'opció de 'Visibilty', es pot habilitar una condició, ja sigui una variable (local o del dispositiu) o una funció que en funció d'un cert valor faci que l'element en concret quan aquesta es compleix deixi de ser visible.



Il·lustració 29: Pestanya que permet condicionar quan no ha de ser visible el rectangle

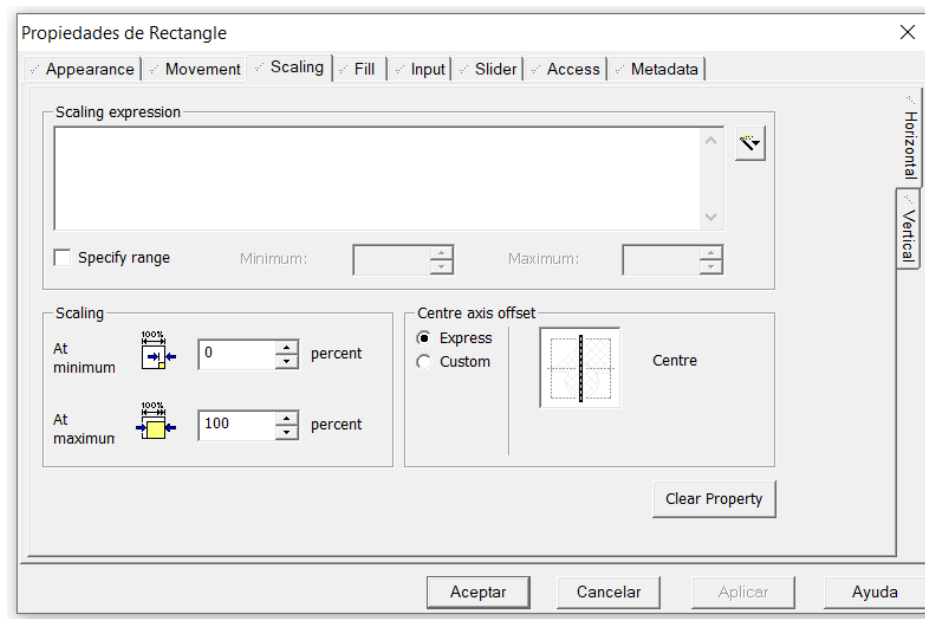
Canviant de pestanya i passant a la de 'Movement', aquí se'ns permet que l'element es desplaci en base al valor de la variable o funció que associem. Especificant un valor mínim (posició inicial) i especificant un valor màxim (posició final), en funció del valor actual de la variable es situarà en un punt d'aquest rang. Es pot configurar perquè és desplaçi tant horitzontal com verticalment.



Il·lustració 30: Configurar si es vol moure l'objecte en funció del valor d'una expressió

La següent opció, 'Scaling', permet que l'amplada d'un objecte canviï dinàmicament en funció d'una variable o funció que associem. Si aquest valor augmenta, l'amplada també ho fa per exemple.

També s'especifica un rang i una equivalència en píxels del desplaçament.



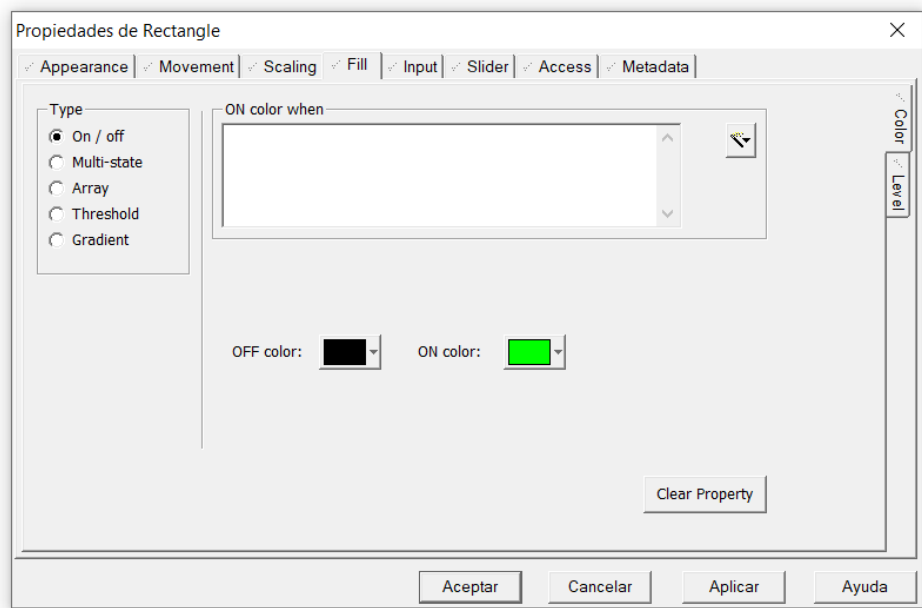
Il·lustració 31: Configuració per determinar l'ample d'un objecte en funció del valor d'una expressió

La pestanya 'Fill' permet que el color interior de l'element variï segons el valor o estat d'una variable o funció.

S'especifiquen cinc casos:

- On/Off: S'habiliten dos colors, un per quan la condició es compleix (ON) i l'altre per quan no (OFF). Simplement, quan la condició (per exemple $A=1$) es compleix, l'interior de l'element tindrà el color corresponent a ON, quan tingui un altre valor, el corresponent a OFF.
- Multi-State: Segueix un principi com l'anterior però seguint el principi d'una taula lògica. En aquest cas es poden arribar a posar fins a cinc condicions diferents i el color interior de l'element varia en funció de quines compleixen la seva condició.
- Array: Referit a un vector de màxim 255 valors, s'hi associa una expressió o variable que retorni cert valor numèric on a cada valor se li correspon un color diferent.

- **Threshold:** El color canvia dinàmicament respecte el valor que se li associa. Especificant quin color ha de tenir l'interior en cada cas. Per exemple, aplicat a un comptador de revolucions, de 0 a 200 el color serà blanc, de 201 a 900 verd i de 901 a 1500 vermell.
- **Gradient:** El color varia gradualment segons va canviant el valor. S'estableix un rang, on en el valor mínim tindrà cert valor, i el valor màxim un altre. La resta de colors aniran canviant automàticament essent un gradient entre els extrems.

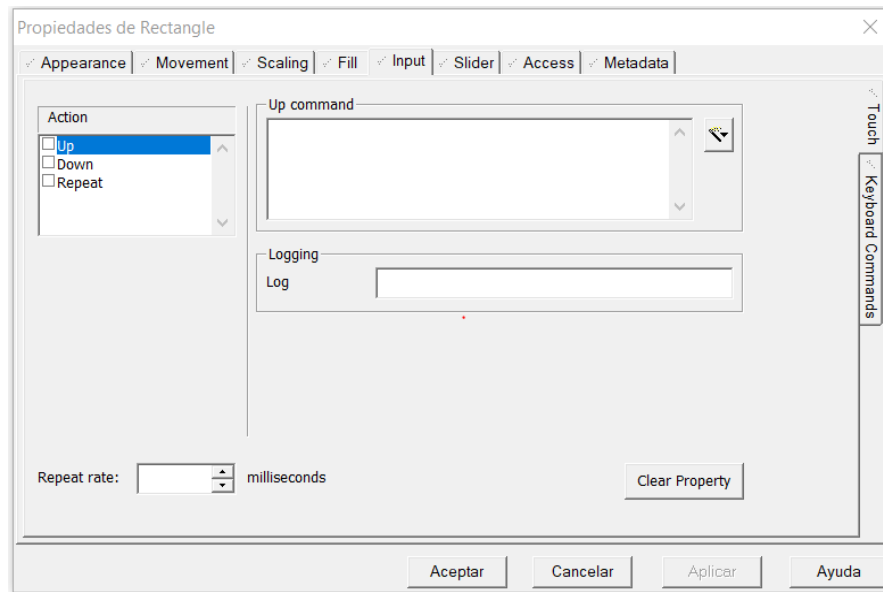


Il·lustració 32: Configurar el color de l'interior d'un objecte en base una certa condició

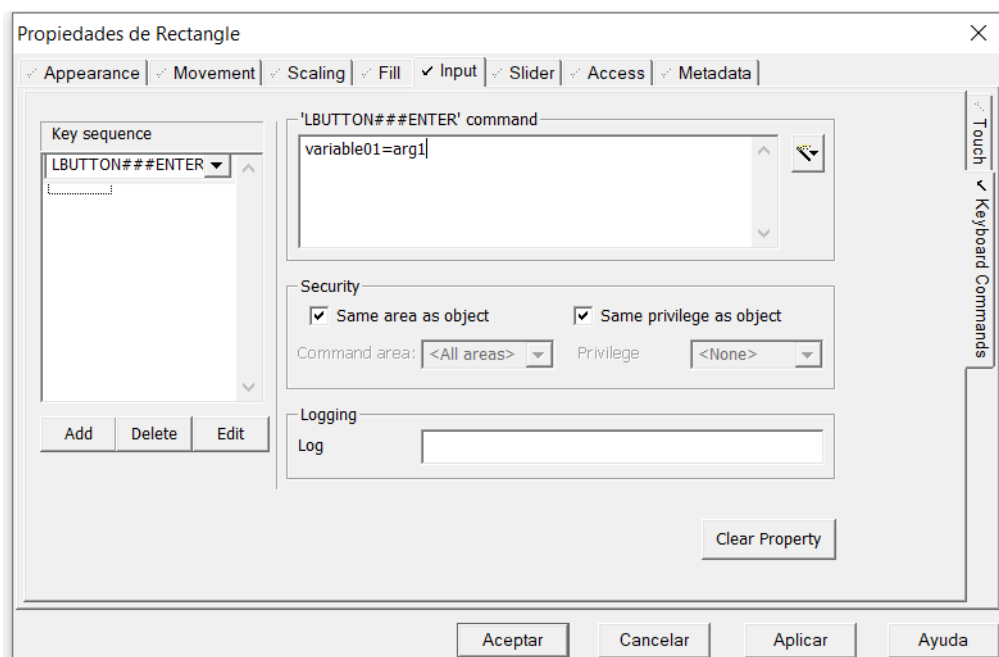
En l'opció 'Input' es permet que s'executi una certa comanda quan es prem l'objecte o s'hi fa una certa comanda del teclat definida prèviament.

Per tant, es pot habilitar que quan es prem el ratolí sobre l'objecte s'executi una funció o que certa variable passi a tenir un valor.

Habilitant-ho des de 'Keyboard Commands', es pot establir una condició perquè s'hi pugui escriure i aleshores que aquest valor passi a la variable associada.



Il·lustració 33: Propietat Input de la configuració



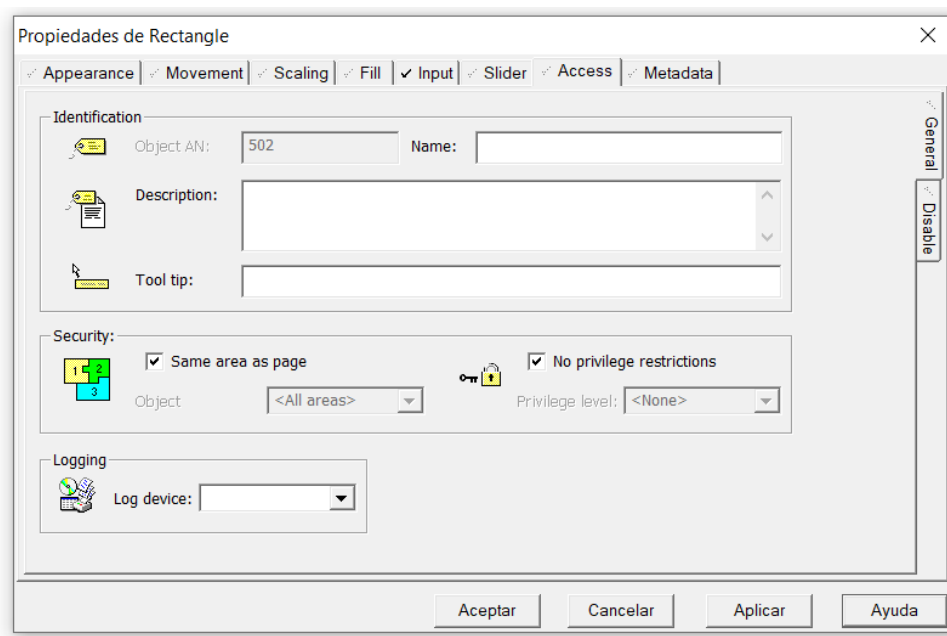
Il·lustració 34: Configuració perquè una variable passi a tenir un nou valor quan es compleix una certa comanda

En la figura X s'observa que aplicant una comanda en el teclat equivalent a: 'Botó Esquerra del ratolí + escriure 3 caràcters + Enter' aquest valor es passarà a la variable anomenada 'variable01'.

En el cas de 'Slider' permet una opció molt semblant a 'Scaling' però aplicant-ho a una barra que es fa més gran o més petita en funció del valor de la variable associada.

La pestanya 'Access' permet configurar quins usuaris poden modificar l'element quan l'SCADA està corrent. Aquesta opció sols té una implicació directa sobre la variable que associem en 'Input'. Sigui quin sigui el rol, podrà visualitzar l'element, però si es configura per certs rols o privilegis, sols aquests hi podran accedir.

També té l'opció aquesta pestanya de poder deshabilitar l'element. Ja no es basa en rols si no, en funció d'una condició, quan aquesta es compleixi, el valor pugui estar ombrejat, deixi de visualitzar-se o simplement que cap usuari pugui tenir-hi accés perquè està totalment deshabilitat.



Il·lustració 35: Configurar quins usuaris tenen accés a l'objecte

Per últim, en 'Metadata' s'hi poden associar una sèrie de noms que es corresponen a uns valors associats a un objecte. Aquesta opció s'utilitza bàsicament entre Genies i Super Genies.

5.3 Genie i Super Genie

Citect i PSO utilitzen principis aplicats en la programació orientada a objectes per facilitar el seu desenvolupament i disseny, ja s'ha vist un clar exemple en el Profile Editor.

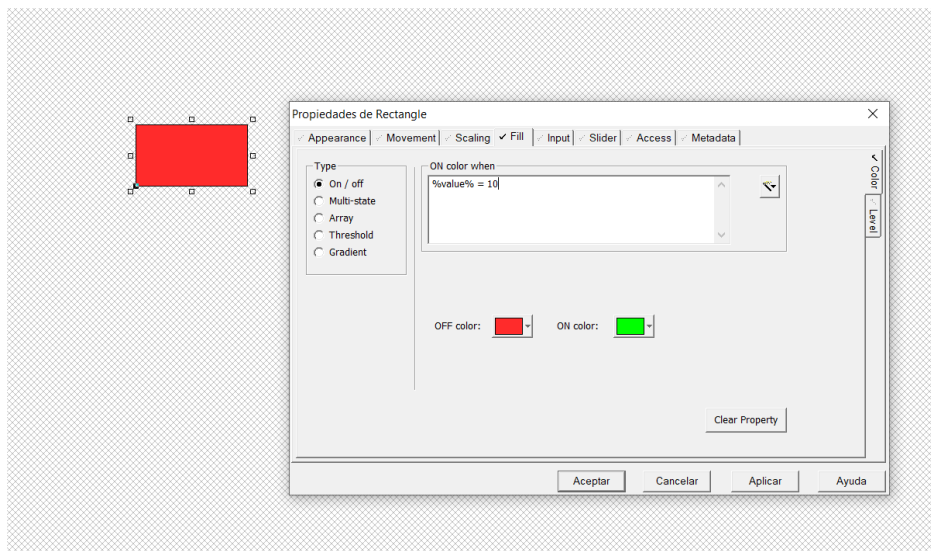
En el desenvolupament gràfic del SCADA fàcilment s'utilitza un mateix element per diferents variables. Un exemple clar seria fer l'SCADA d'una planta industrial i mostrar l'estat de diferents motors. No és òptim haver de dissenyar cada motor individualment, i optar per una opció de copiar i enganxar tampoc és la més adequada.

En aquesta situació, el que es fa és crear un Genie i un Super Genie.

El Genie seria per crear el dibuix del motor i a aquest s'hi associaria un Super Genie que per exemple, seria un pop-up que permetés engegar i apagar el motor quan es premés l'element.

Per crear un nou Genie cal fer-ho des de 'File' i 'New', seleccionant l'opció de Genie.

L'element o conjunt a visualitzar es dissenyen igualment que com s'ha explicat anteriorment. La única diferència és que no s'associa aquí cap variable. El que es vol és un element que es permeti replicar-lo fàcilment, on per exemple, si es vol que el seu interior es posi vermell quan el valor de la intensitat superi els 10A i es tenen els motors A, B i C amb les senyals de corrent Ia, Ib i Ic, es fa doncs, un cas genèric en el Genie en la pestanya de 'Fill'. Així s'associa una variable general que quan sigui 10 canviï de color. Aquesta, en el Genie, ha d'anar representada entre '%'. Per tant, l'expressió seria: "`%value% = 10`".

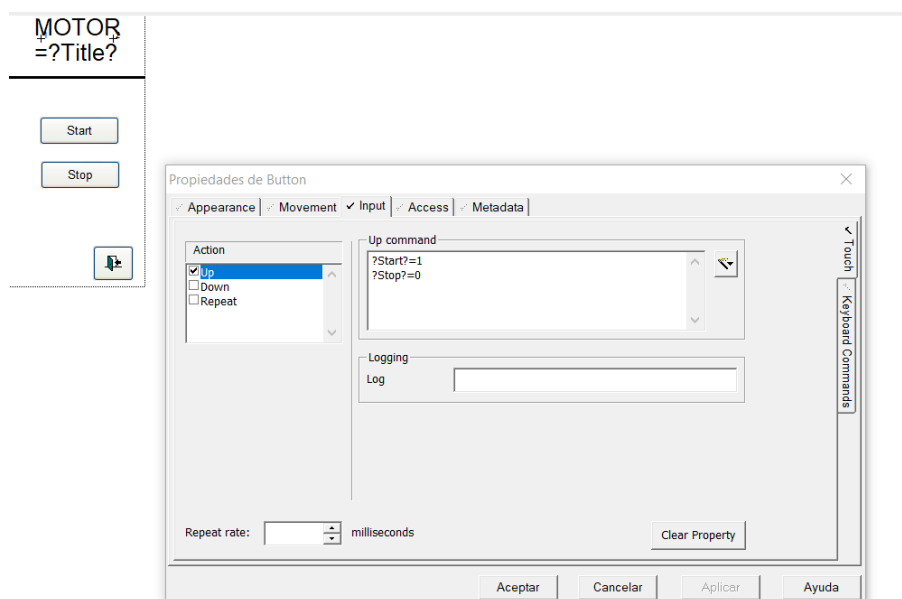


Il·lustració 36: Configurar el Genie perquè es posi en verd quan la variable que s'associarà a ell valgui 10

Així, quan el Genie s'utilitzi en les pantalles haurem d'establir quina variable assignem a 'value' en cada cas únicament.

Tan s'hi poden correspondre variables, funcions, texts (strings) o valors numèrics.

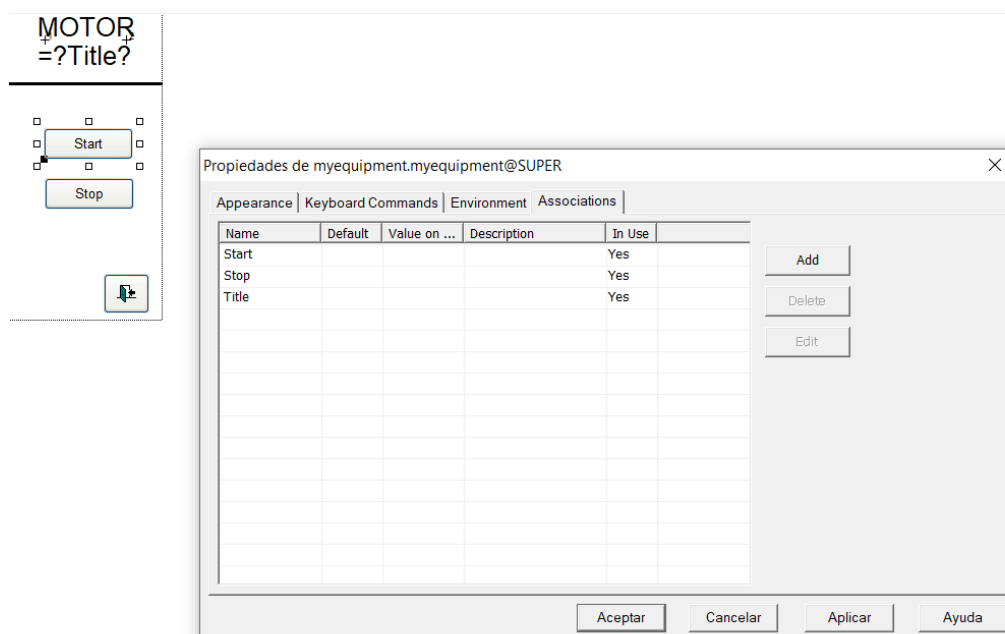
El Super Genie es crea d'igual manera. La única diferència es que l'expressió enlloc d'anar entre '% ' va entre ' ? '.



Il·lustració 37: Configuració del Super Genie quan es prem el botó 'Start'

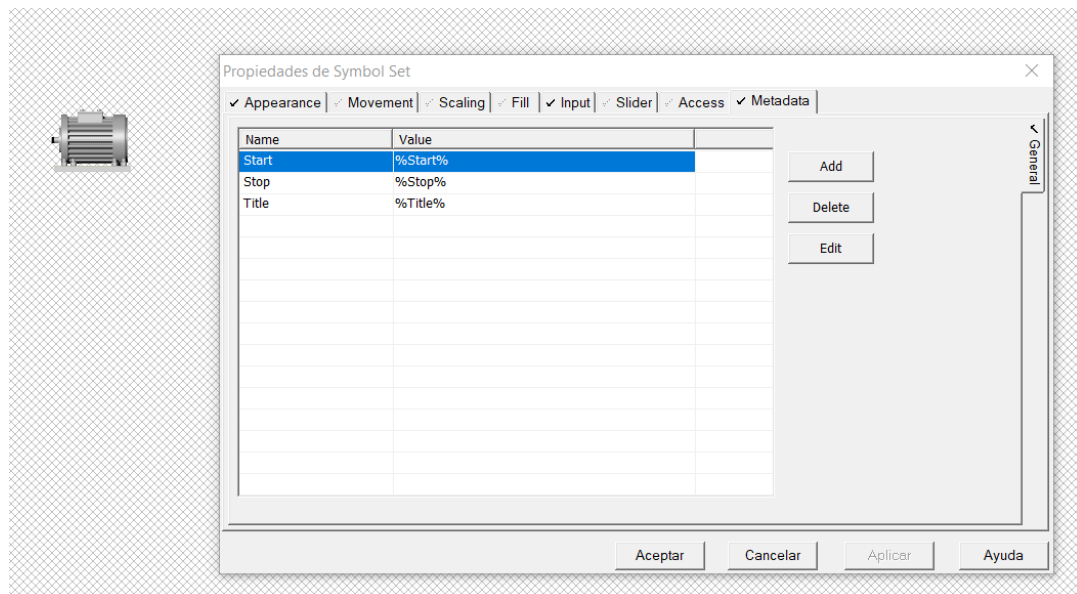
El que relaciona el Genie i el Super Genie és la pestanya de 'Metada' del Genie.

Cal primer que en Super Genie, en 'Page Properties' a Association s'hi relacionin els diferents elements. Aquests surten pràcticament automàticament sols prement-hi.



Il·lustració 38: Configurar en les propietats del Super Genie l'associació de variables

En el Genie, a Metada, cal que les variables que s'utilitzin en el Genie i el Super Genie tinguin el mateix nom.



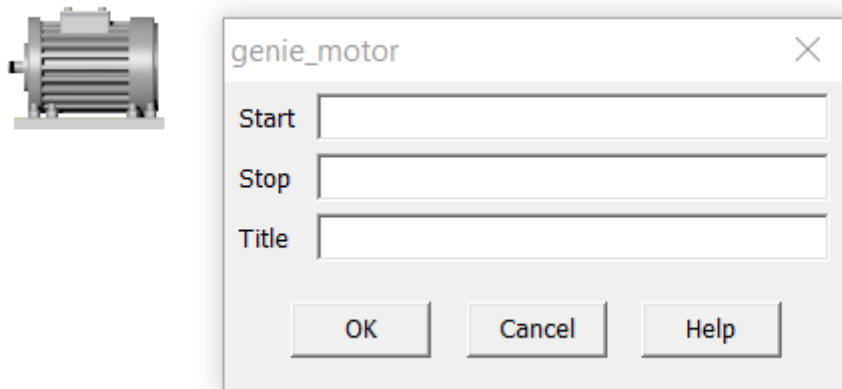
Il·lustració 39: Relació entre les variables del Genie i les del Super Genie

Perquè s'obri el Super Genie, com a pop-up per exemple, al prémer el Genie quan es visualitzi l'SCADA, cal que en la pestanya 'Input' del Genie s'hagin cridat les següents funcions:

```
AssMetadata(-2,-2);
```

```
AssPopUp("nom del Super Genie" , "Clúster")
```

Cal dir que no és obligatori assignar un Super Genie a un Genie, tot depèn de l'aplicació que es necessiti.



Il·lustració 40: Finestra que s'obra en la pantalla del SCADA per configurar els valors del Genie

En la figura X s'observa com el Genie s'ha posat en una de les pàgines de l'SCADA i com es s'hi associen les variables en aquest. D'aquesta manera es simplifica molt el procés on, configurant i fent modificacions en un únic lloc, es veu reflectit en tots els que hi ha en el projecte.

6. Programació Cicode

Com a qualsevol SCADA és necessari utilitzar funcions de programació amb la fi de realitzar operacions, canviar estat de variables en funció del que pugui estar succeint en l'SCADA o en els dispositius a que està connectat.

En aquest cas, el PSO treballa amb una aplicació anomenada Cicode Editor que permet escriure les diferents funcions a implementar en el programa.

Es poden utilitzar dos tipus de llenguatge, Visual Basic o Cicode. És recomanable utilitzar el Cicode per tal com està pensat el programa i és com estan escrites les funcions que duu el programa per defecte. Aquestes últimes es poden utilitzar igualment en els projectes.

El llenguatge Cicode és semblant Al llenguatge C de programació amb petites variacions.

D'entrada accepta els únics tipus de dades que s'hi poden declarar són els següents:

- INT: Fins a 32 bits, únicament valors enters, s'hi inclouen valors digitals.
- REAL: Fins a 64 bits, punt flotant doncs, les variables amb decimals (reals) són d'aquest tipus.
- STRING: Cadena de text de fins a 128 bytes.

- Object: Pertanyent als objectes de control ActiveX del Citect.

S'hi poden declarar variables Globals. Declarant-les en un únic arxiu serveixen per tot el projecte ja i no únicament per l'arxiu on estan.

Les funcions poden executar una o varies operacions, o retornar un valor segons com es declarin.

- Funció que realitza una o més operacions:

```
FUNCTION "Nom de la funció"()  
[...]  
END
```

- Funció que retorna un valor:

```
"Tipus de dada a retornar" FUNCTION "Nom de la funció"()  
[...]  
RETURN "variable";  
END
```

S'hi poden aplicar condicions i bucles com qualsevol altre llenguatge de programació.

Es pot organitzar la programació en arxius '.ci' si es fa amb Cicode. No hi ha un programa principal, si no que les funcions es van cridant individualment i segons necessitat en els elements del SCADA o entre funcions. Per tant, tots els arxius i funcions d'un mateix projecte estan enllaçats, per tant la distribució per arxius és un mer tema organitzatiu de cada desenvolupador.

7. Power SCADA Operation Help

Al instal·lar el PSO també es descarrega un arxiu, amb el nom d'aquest capítol, que s'obra amb el buscador però que funciona sense connexió el qual és una guia pel programa.

Qualsevol tema relacionat amb el PSO està allà, és doncs una guia, la qual és molt útil per entendre i aprofundir més amb el software i per tant és un gran recurs amb el qual és important treballar-hi.

ANNEX 3

MANUAL D'ÚS DE MICROSOFT AZURE

SUMARI

| | |
|---|----|
| Índex de Figures | 2 |
| 1. Objecte del document..... | 3 |
| 2. IoT Hub | 3 |
| 2.1 Crear i configurar un nou recurs IoT Hub..... | 3 |
| 2.2 Treballar amb IoT Hub..... | 6 |
| 3. Azure SQL Database | 7 |
| 3.1 Crear i configurar una nova base de dades Azure SQL | 7 |
| 3.2 Treballar amb Azure SQL..... | 10 |
| 4. Stream Analytics..... | 12 |
| 4.1 Crear i configurar un nou Stream Analytics | 12 |

Índex de Figures

| | |
|---|----|
| Il·lustració 1: Mostra de diferents serveis d'Azure..... | 3 |
| Il·lustració 2: Cerca entre els recursos d'Azure i selecció d'IoT Hub | 4 |
| Il·lustració 3: Configuració d'IoT Hub | 4 |
| Il·lustració 4: Pantalla principal d'Azure IoT Hub | 5 |
| Il·lustració 5: Claus d'accés de l'IoT Hub creat | 6 |
| Il·lustració 6: Dispositiu connectat amb l'IoT Hub | 6 |
| Il·lustració 7: Gràfics que indiquen l'històric de missatges rebuts en el recurs..... | 7 |
| Il·lustració 8: Cerca i selecció per crear el recurs SQL Database..... | 7 |
| Il·lustració 9: Configuració bàsica de la base de dades | 8 |
| Il·lustració 10: Configuració de la xarxa i connexió de la base de dades | 9 |
| Il·lustració 11: Configuració addicional de la base de dades | 10 |
| Il·lustració 12: Portal principal del servidor de la base de dades..... | 11 |
| Il·lustració 13: Pantalla principal de la base de dades | 11 |
| Il·lustració 14: Selecció per poder crear un nou recurs Stream Analytics | 12 |
| Il·lustració 15: Configuració del Stream Analytics..... | 13 |
| Il·lustració 16: Pantalla principal de Stream Analytics | 13 |

1. Objecte del document

La motivació d'aquest document es servir de referència bàsica per l'aprenentatge i ús del Power SCADA Operation, ja que existeix escassa informació o exemples sobre com utilitzar-lo.

D'aquesta manera, amb aquest manual es comprendrà amb més facilitat el programa i com crear un projecte bàsic amb aquest.

2. IoT Hub

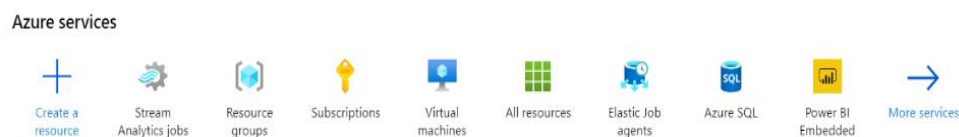
2.1 Crear i configurar un nou recurs IoT Hub

Des d'Azure s'ofereixen diferents camins a seguir amb la fi de crear un nou recurs IoT Hub. Aquests inclouen per exemple, fer-ho des del portal principal d'Azure, des del Power Shell de Window o des del CLI d'Azure.

En aquest cas, s'opta per fer-ho des del portal principal d'Azure, ja que és el que permet configurar-ho d'una forma més visual i intuïtiva, permetent incidir en més àmbits més fàcilment.

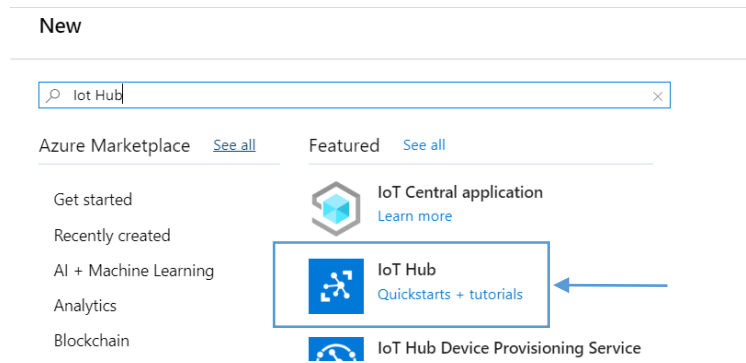
A continuació, s'explica per passos el procés a seguir. Cal però, tenir en compte que abans s'ha d'haver creat una compta i/o subscripció a Azure.

1. Accedir al Portal Principal d'Azure i entrar a "Create a resource".



Il·lustració 1: Mostra de diferents serveis d'Azure

2. Buscar en el cercador que apareix "IoT Hub" i accedir-hi.



Il·lustració 2: Cerca entre els recursos d'Azure i selecció d'IoT Hub

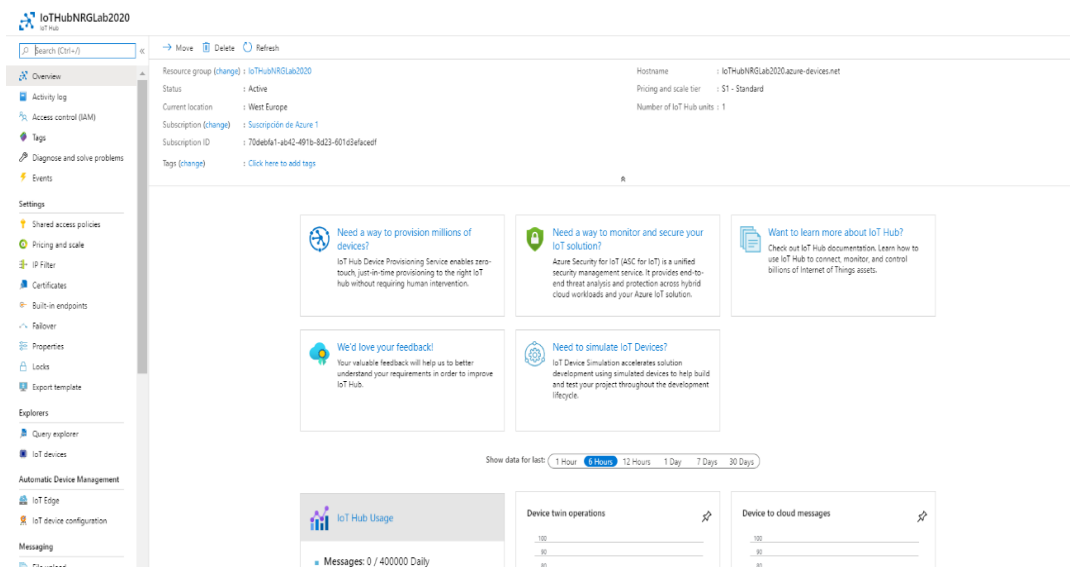
3. Ens apareix una nova finestra on hem de realitzar la configuració bàsica del nou IoT Hub que creem.

Il·lustració 3: Configuració d'IoT Hub

4. En aquest cas, cal que completeu tots els camps.
 - Grup de recursos: Si en tenim un de creat, el seleccionem, si no, en creem un.

- **Regió:** Hem d'elegir la ubicació del centre de dades on volem allotjar el nostre recurs. La forma correcta seria localitzar tot el grup de recursos en la mateixa ubicació, i també que aquesta estigui el més pròxima possible.
També hem de tenir en compte les prestacions que ens ofereix cada cas, per si algun dels recursos que volem utilitzar no està disponible en certa regió.
A Espanya no n'hi ha cap per ara (està previst fer-ne un), en aquest cas doncs seleccionem Europa Oest, localitzat als Països Baixos.
- **Nom de l'IoT Hub:** Creem un nom, s'ha de tenir en compte que estigui disponible.
- **Nivell de Preu:** Tenim l'opció de seleccionar el tipus de quota a seleccionar segons les necessitats que tinguem en la nostra aplicació. Podent elegir entre:
 - Estàndard: Preu escalable en funció de la quantitat de missatges diaris a enviar. Té totes les característiques possibles.
 - Bàsica: Mateixa metodologia que en Estàndard però té moltes menys característiques disponibles.
 - Gratuïta: Equival a una quota estàndard però es té la limitació de no poder enviar més de 8.000 missatges diaris i una mida de kB més reduïda per missatge.
- **Azure Security Center:** A elecció pròpia si és vol contractar, implica un sobrecost.

5. Seguidament ja podem crear el recurs. Ens apareixerà després la pantalla principal d'aquest.

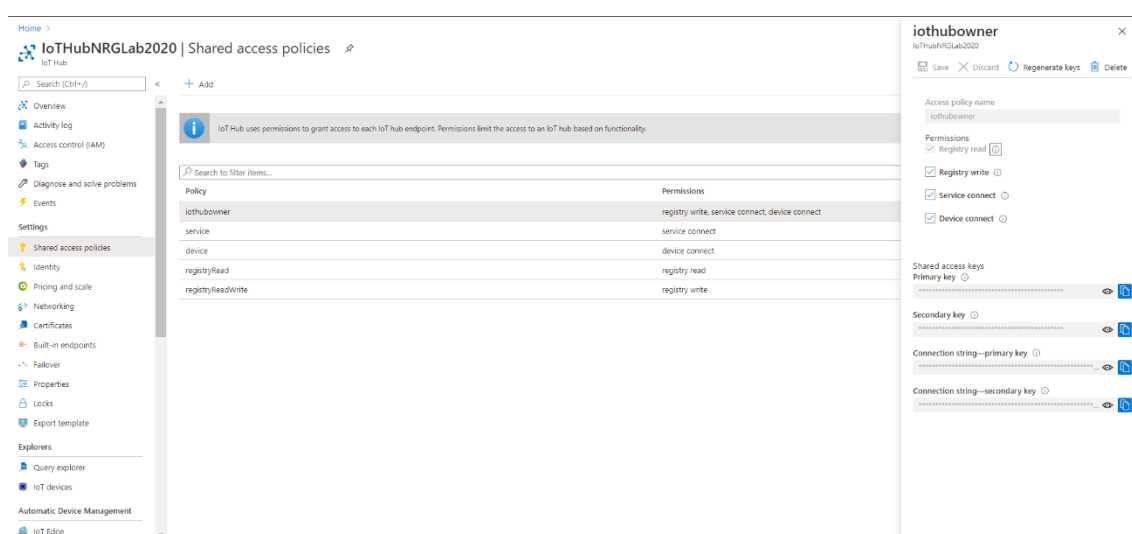


Il·lustració 4: Pantalla principal d'Azure IoT Hub

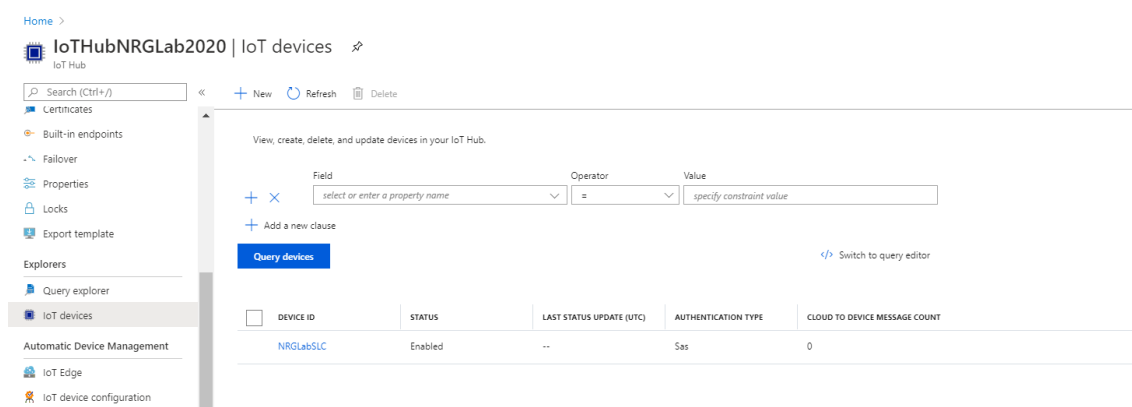
2.2 Treballar amb IoT Hub

Des de la pantalla principal d'Azure IoT Hub podem accedir a les diferents configuracions i característiques que ofereix.

Des de les 'Directives d'accés compartit' s'accedeix a les diferents claus que permeten l'accés des de dispositius a l'IoT Hub creat per l'usuari, on es destaca la clau principal de connexió com a STRING, que és l'element que necessita el dispositiu per poder-se connectar.



Il·lustració 5: Claus d'accés de l'IoT Hub creat

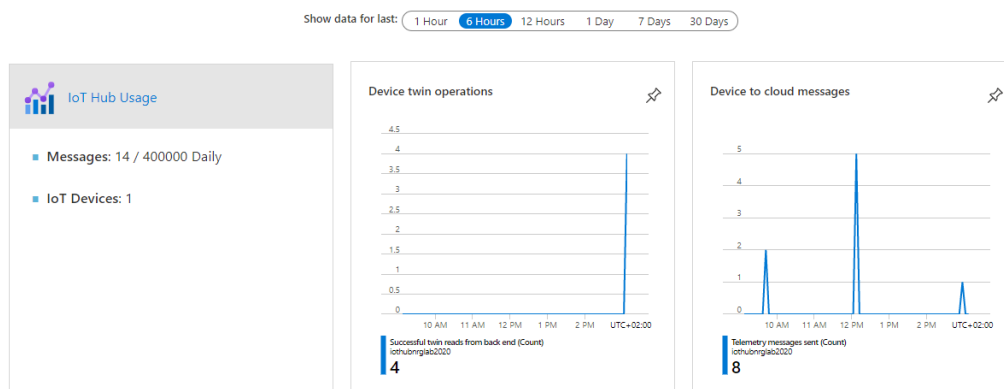


Il·lustració 6: Dispositiu connectat amb l'IoT Hub

En l'explorador, a 'Dispositius IoT', trobem el llistat de dispositius que es connecten amb l'IoT Hub i on es poden editar individualment.

En el panell principal podem monitorar si s'estan enviant i rebent correctament els missatges, apareixent en una gràfica la quantitat de missatges que IoT Hub està rebent. No es pot però,

saber el contingut d'aquests des de l'IoT Hub ja que aquest servei és únicament una central de missatgeria.



Il·lustració 7: Gràfics que indiquen l'històric de missatges rebuts en el recurs

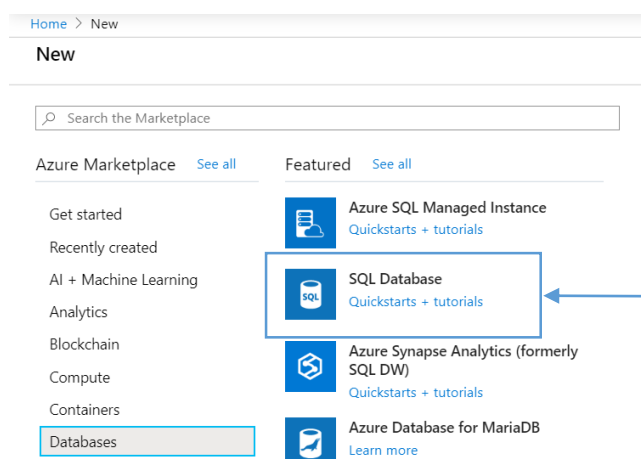
3. Azure SQL Database

3.1 Crear i configurar una nova base de dades Azure SQL

Per crear un nou recurs Azure SQL DataBase es farà des del portal principal d'Azure, el qual permet poder-ho configurar més fàcilment.

El procés que es seguirà implica crear un nou servidor de base de dades SQL i allà allotjar-ne una perquè el procés és conjunt.

1. Accedir al Portal d'Azure i entrar a "Create a resource".
2. Buscar en el cercador "SQL Database".



Il·lustració 8: Cerca i selecció per crear el recurs SQL Database

3. Ens apareix una nova finestra on hem de fer la configuració bàsica de la BD:

Create SQL Database
Microsoft

Basics Networking Additional settings Tags Review + create

Create a SQL database with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. [Learn more](#)

Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Database details
Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name *

Server * [Create new](#)

Want to use SQL elastic pool? * ☐ Yes ☒ No

Compute + storage * **General Purpose**
Gen5, 2 vCores, 32 GB storage
[Configure database](#)

II-l·lustració 9: Configuració bàsica de la base de dades

- **Grup de Recursos:** Busquem el grup de recursos on tenim els altres recursos creats i volem agrupar, tals com l'IoT Hub.
- **Nom de la BD:** Posem un nom per la base de dades.
- **Servidor:** Posem el nom del servidor a utilitzar, si n'hem de crear un nou. Apareixerà doncs en el lateral uns camp a omplir.
 - Nom del Servidor: Ha de ser amb minúscules.
 - Nom d'entrada d'administrador: Creem un nom d'usuari per l'administrador.
 - Contrasenya: Creem una contrasenya per poder accedir-hi.
 - Localització: Localitzem el servidor, si és possible, en la mateixa regió que la resta del grup de recursos.
- **Ús de SQL Elastic Pool:** Sols si ens interessa acceptem.
- **Configuració d'espai d'emmagatzematge i capacitat de computació:** Configurem la nostra base de dades en funció de les necessitats que tinguem. Hem de tenir en compte les DTUs i l'espai que es necessiti.

4. Passem a la següent pestanya on configurarem la xarxa i connexió. Per simplificar el procés i evitar conflictes de connectivitat, el més fàcil es deixar-ho tal com la imatge següent.

Basics **Networking** Additional settings Tags Review + create

Configure network access and connectivity for your server. The configuration selected below will apply to the selected server 'rogerprova1' and all databases it manages. [Learn more](#)

Network connectivity

Choose an option for configuring connectivity to your server via public endpoint or private endpoint. Choosing no access creates with defaults and you can configure connection method after server creation. [Learn more](#)

Connectivity method * ⓘ

☐ No access

☒ Public endpoint

☐ Private endpoint

Firewall rules

Setting 'Allow Azure services and resources to access this server' to Yes allows communications from all resources inside the Azure boundary, that may or may not be part of your subscription. [Learn more](#)

Setting 'Add current client IP address' to Yes will add an entry for your client IP address to the server firewall.

Allow Azure services and resources to access this server *

No Yes

Add current client IP address *

No Yes

Il·lustració 10: Configuració de la xarxa i connexió de la base de dades

5. Passant a la següent pestanya, Configuració addicional, determinem:
- Si volem començar amb una base de dades buida o volem introduir inicialment dades d'una altra base o d'un altre lloc.
 - La intercalació, fent referència a les normes d'ordre i comparació de les dades. La configuració que s'estableix ja no podrà ser modificada en un futur. Per defecte es dóna: SQL_Latin1_General_CP1_CI_AS que implica:
 - Latin1_General: Tipus de caràcters de llenguatge ASCII.
 - CP1: Tipus de codificació, referent a la pàgina CP-1252 de Windows, on s'inclou la codificació de caràcters.
 - CI: La no distinció entre lletres majúscules i minúscules.
 - AS: Existència i comprensió dels accents i altres signes diacrítics(ü, ú o u no s'interpretaran igual en la base de dades).

Create SQL Database
Microsoft

Basics Networking **Additional settings** Tags Review + create

Customize additional configuration parameters including collation & sample data.

Data source

Start with a blank database, restore from a backup or select sample data to populate your new database.

Use existing data * None Backup Sample

Database collation

Database collation defines the rules that sort and compare data, and cannot be changed after database creation. The default database collation is SQL_Latin1_General_CP1_CI_AS. [Learn more](#)

Collation * ⓘ SQL_Latin1_General_CP1_CI_AS
[Find a collation](#)

Advanced data security

Protect your data using advanced data security, a unified security package including data classification, vulnerability assessment and advanced threat protection for your server. [Learn more](#)

Get started with a 30 day free trial period, and then 12.6495 EUR/server/month.

Enable advanced data security * ⓘ Start free trial Not now

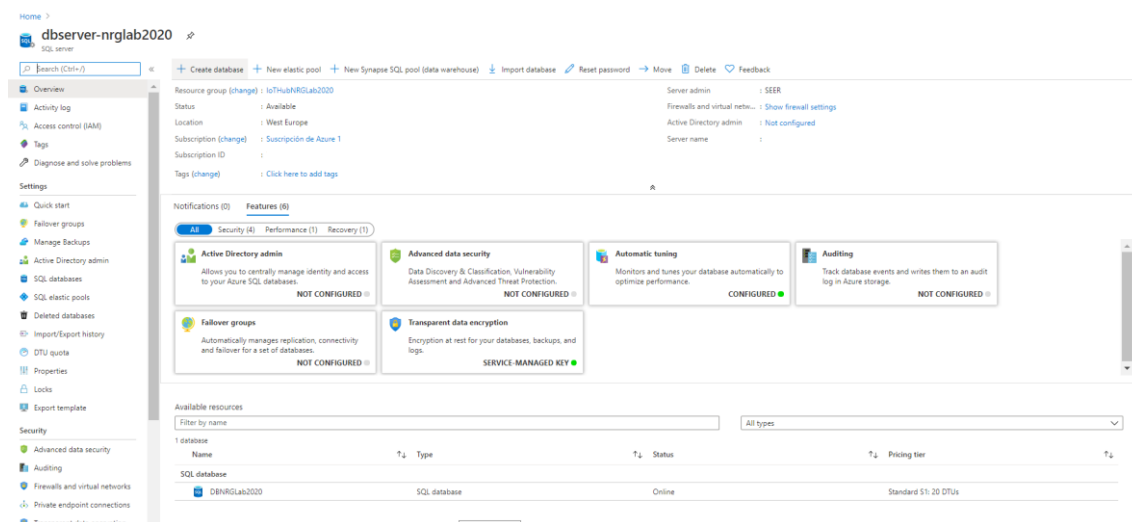
Advanced data security will automatically create a new storage account for saving vulnerability assessments. If a storage account was previously created for this purpose, it will be used instead. Azure storage prices will apply.

Il·lustració 11: Configuració adicional de la base de dades

- Finalment ja es pot crear la base de dades. Des d'aquesta s'han de configurar les diferents IP's en el tallafocs (Firewall) per poder-hi tenir accés. També es pot obrir la base de dades d'Azure localment des del Visual Studio.

3.2 Treballar amb Azure SQL

Quan es crea un nou servidor SQL, hi ha també la base de dades que s'ha creat. S'hi poden crear altres bases de dades en el servidor, al cap i a la fi funciona com un servidor de base de dades Microsoft SQL local però en aquest cas està en el núvol.



Il·lustració 12: Portal principal del servidor de la base de dades

El servidor inclou un 'Nom de servidor' que és una adreça web que permet poder accedir-hi des d'altres programes tals com Visual Studio o aplicacions de control en temps real.

Accedint a la base de dades, tenim en la pantalla un recull de les principals característiques i configuracions d'aquesta. Des d'aquí tenim l'opció d'accedir al talla focs d'aquesta i configurar les IP's que tenen accés a la base de dades. Si es vol accedir a aquesta i fer-hi alguna consulta o modificació, o si es vol que un dispositiu o aplicació hi pugui enviar dades o llegir-les, cal que la IP associada aquest estigui declarada com a client en el talla focs.



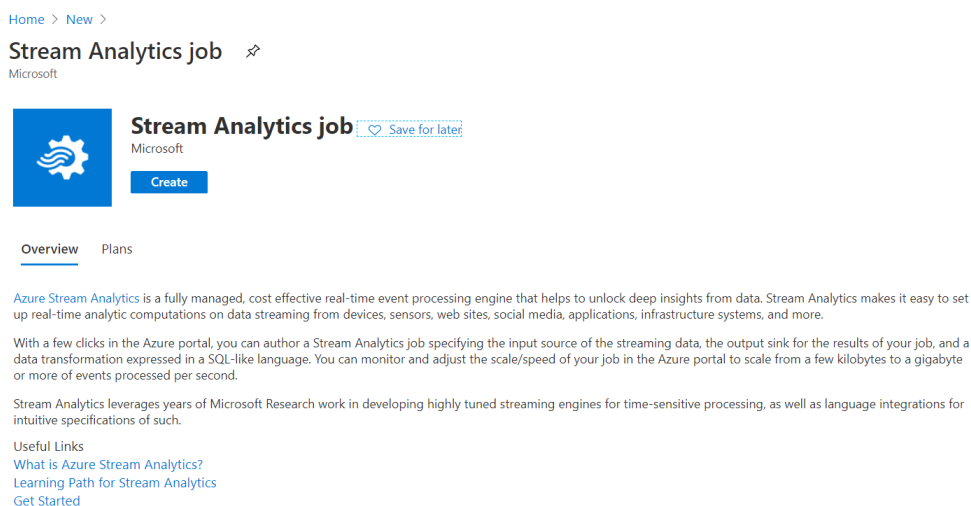
Il·lustració 13: Pantalla principal de la base de dades

En aquesta pantalla principal apareix unes gràfiques que informen sobre l'històric de del percentatge d'ús de computació, és a dir, de les DTUs; alhora un altre gràfic expressa l'ús d'espai d'emmagatzematge que resta disponible en la base de dades.

Accedint a l'opció 'Editor de Query', permet entrar en la base de dades i poder veure i modificar el contingut de les diferents taules tal com es podria fer obrint la base de dades en el Visual Studio per exemple.

4. Stream Analytics

4.1 Crear i configurar un nou Stream Analytics



Il·lustració 14: Selecció per poder crear un nou recurs Stream Analytics

Per crear un nou servei Azure Stream Analytics cal anar al portal principal d'Azure i cercar-lo.

Apareixerà un nova pantalla on cal anomenar-lo i associar-lo a un grup de recursos, o crear-ne un si no s'ha fet abans.

Cal seguir el principi indicat en anteriors capítols i intentar situar aquest servei on estan la resta d'elements del grup de recursos.

Si es vol que el servei es desenvolupi en el núvol cal elegir l'opció de 'Cloud' en la configuració de 'Hosting Enviroment'. Pel cas contrari, si es vol realitzar en una passarel·la (en anglès gateway) d'un dispositiu IoT cal seleccionar l'opció 'Edge'.

Finalment, cal seleccionar les unitats de computació, que vindria a ser la quantitat de CPU que es requerirà. Tot depèn de quines i quantes dades seran processades per l'Stream Analytics.

New Stream Analytics job

i This will create a new Stream Analytics job. You will be charged according to Azure Stream Analytics billing model. [Learn more.](#) →

Job name *

Subscription *

Resource group *

Location *

Hosting environment ⓘ

Streaming units (1 to 192) ⓘ

Create

Il·lustració 15: Configuració del Stream Analytics

Un cop creat, s'accedeix a la pantalla principal d'aquest. Cal aleshores, configurar cada una de les entrades i sortides que es volen en l'Stream Analytics.

Les sortides, si s'associen a una base de dades, cal que cada una correspongui a una taula diferent, encara que totes pertanyin a una mateixa base de dades, ja que cada sortida es correspon a una taula en aquest cas.

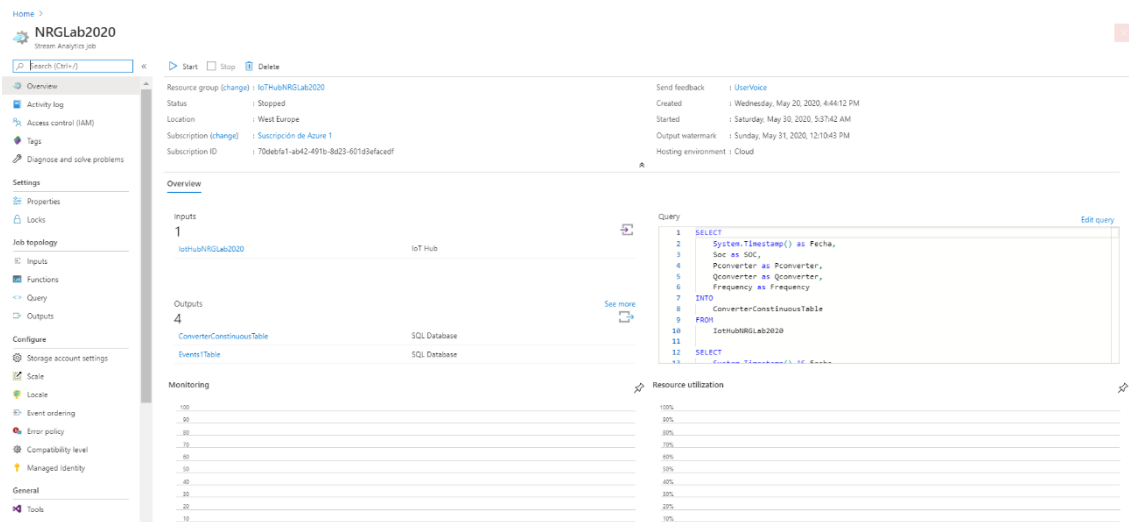
Un cop fet això, sols cal accedir a l'editor de querys i escriure les senzilles querys que permetin passar les dades des de l'entrada cap a la sortida.

Un exemple en pot ser:

```
SELECT
    System.Timestamp() as Fecha,
    Soc as SOC,
    Pconverter as Pconverter,
    Qconverter as Qconverter,
    Frequency as Frequency
INTO
    ConverterConstinuousTable
FROM
    IotHubNRGLab2020
```

Una última característica a configurar és la ‘Política d’Errors’ on cal indicar què ha de fer el sistema si no envia correctament una dada i sorgeix un error o avís. Per defecte apareix l’opció de ‘Reintentar’, on el servei seguirà reintentant enviar el missatge fins que ho aconsegueixi o simplement, després de que en el primer intent l’enviament no hagi estat fructífer abandona aquest i passa al següent.

És recomanable elegir la segona opció per evitar problemes en l’enviament de dades.



Il·lustració 16: Pantalla principal de Stream Analytics

Per iniciar la comunicació entre entrades i sortides cal prémer a ‘Start’. Aquesta no s’aturarà fins que es premi ‘Stop’. Per tant, es pot tancar l’ordinador o pot apagar-se el dispositiu i no estar enviant dades a IoT Hub però l’Stream Analytics seguirà treballant. Cal tenir en compte aleshores que el cost va associat a les hores d’ús que se’n fa d’aquest i de les unitats de computació contractades.

ANNEX 4

INFORME AMB POWER BI



INFORME DIARIO DE OPERACIÓN DEL BESS

NRG.Lab

ÍNDICE

| | |
|-----------------------|---|
| INTRODUCCIÓN | 2 |
| ESTUDIO DE POTENCIAS | 3 |
| ESTUDIO DE TENSIONES | 4 |
| ESTUDIO DE CORRIENTES | 5 |

INTRODUCCIÓN

El objetivo de este documento es mostrar un informe sobre el funcionamiento del BESS a lo largo de un día.

Se plantea un estudio y análisis de las diferentes señales de interés obtenidas, para así poder visualizar la operación del sistema i poder detectar cualquier comportamiento anómalo.

Para autocompletar el documento, es necesario indicar qué fecha se desea visualizar en el informe.

miércoles, 3 de junio de 2020

Selección Fecha para Potencias

2020 (Año) + junio (Mes) ... ✓

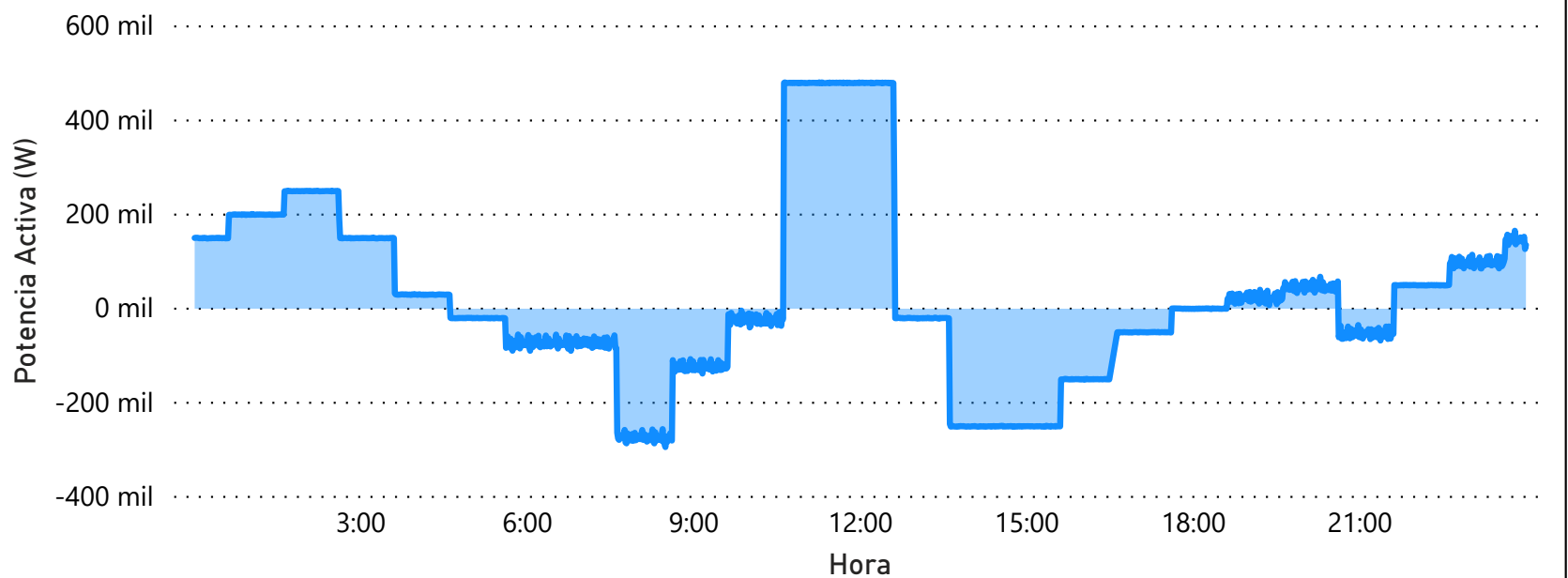
Selección Fecha para Tensiones y Corrientes

2020 (Año) + junio (Mes) ... ✓

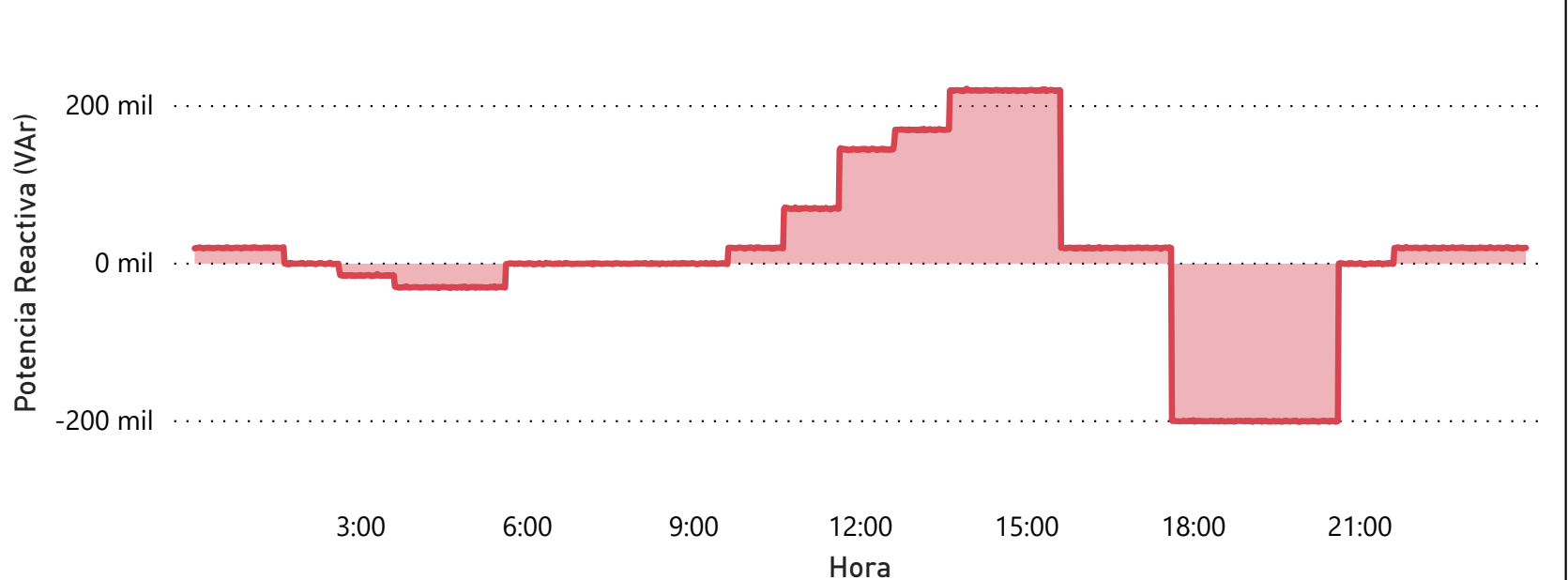
ESTUDIO DE POTENCIAS

Los datos obtenidos durante el miércoles, 3 de junio de 2020 indican que los valores de potencia máximos en la salida del BESS han sido de 481,09 mil W y 222,17 mil VAR, y los valores mínimos han sido de -294,05 mil W y -200,92 mil VAR, respectivamente. Siendo los valores promedios diarios de 25,30 mil W y 12,84 mil VAR.

Potencia Activa



Potencia Reactiva



ESTUDIO DE TENSIONES

Los datos obtenidos durante el miércoles, 3 de junio de 2020 indican que los valores promedio de tensión en la salida del convertidor, en Voltios, han sido:

Promedio de Va_LV



207,41

Promedio de Vb_LV



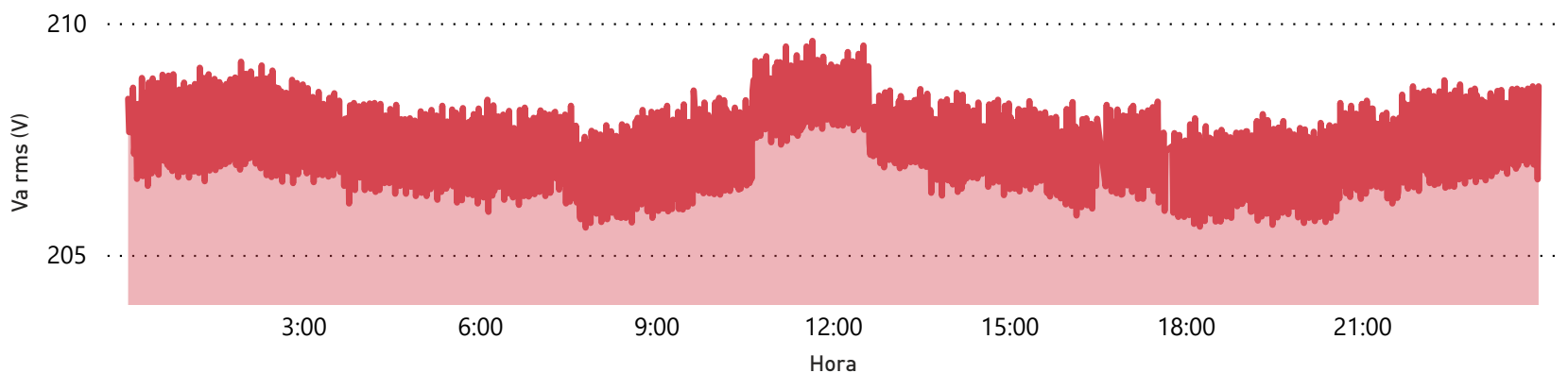
207,51

Promedio de Vc_LV

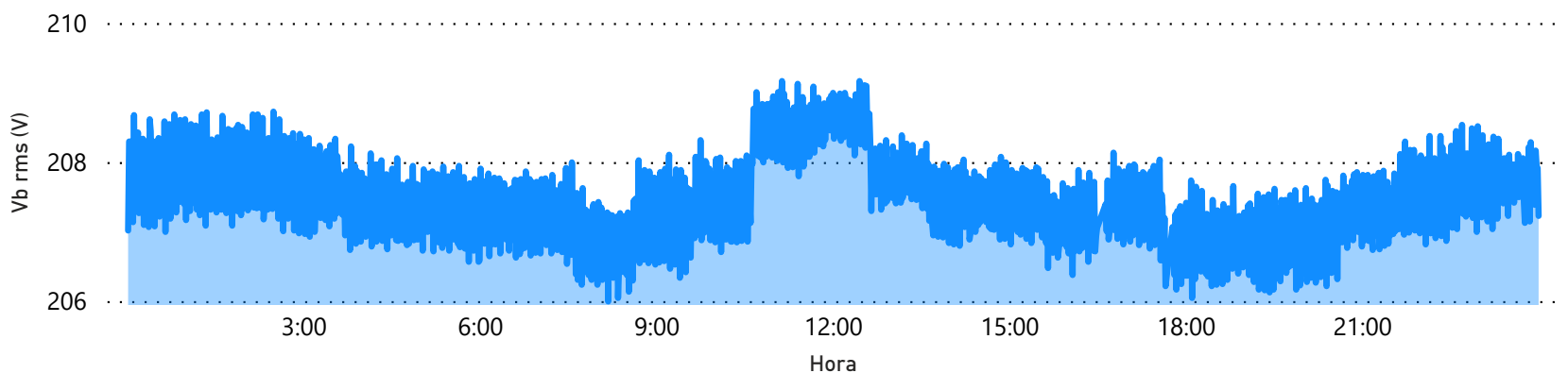


208,49

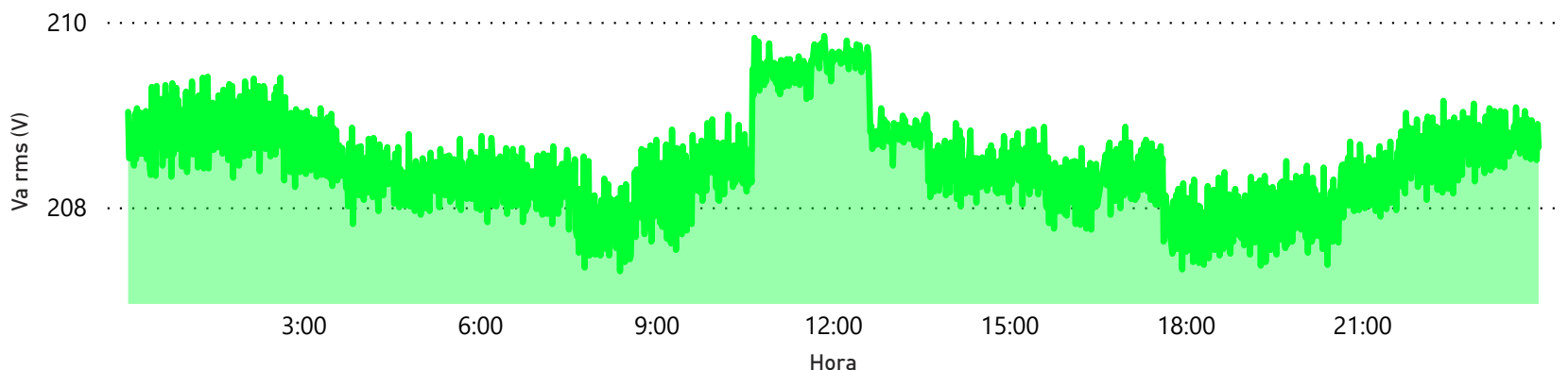
Tensión eficaz fase a



Tensión eficaz fase b



Tensión eficaz fase c



ESTUDIO DE CORRIENTES

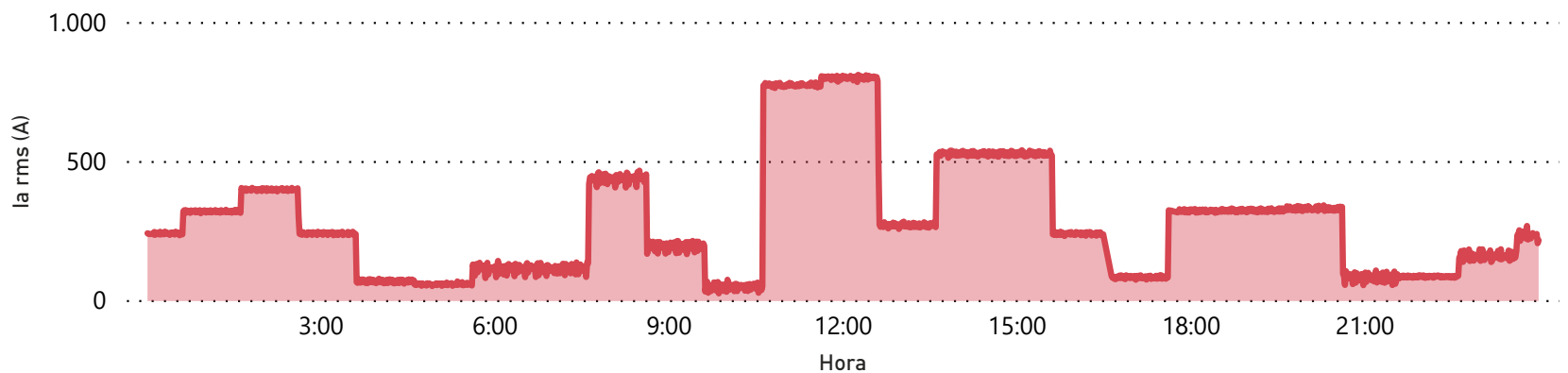
Los datos obtenidos durante el miércoles, 3 de junio de 2020 indican que los valores de las corrientes, en Amperios, han sido:

| Máx. de Ia_LV | Mín. de Ia_LV |
|---------------|---------------|
| 813,50 | 27,50 |

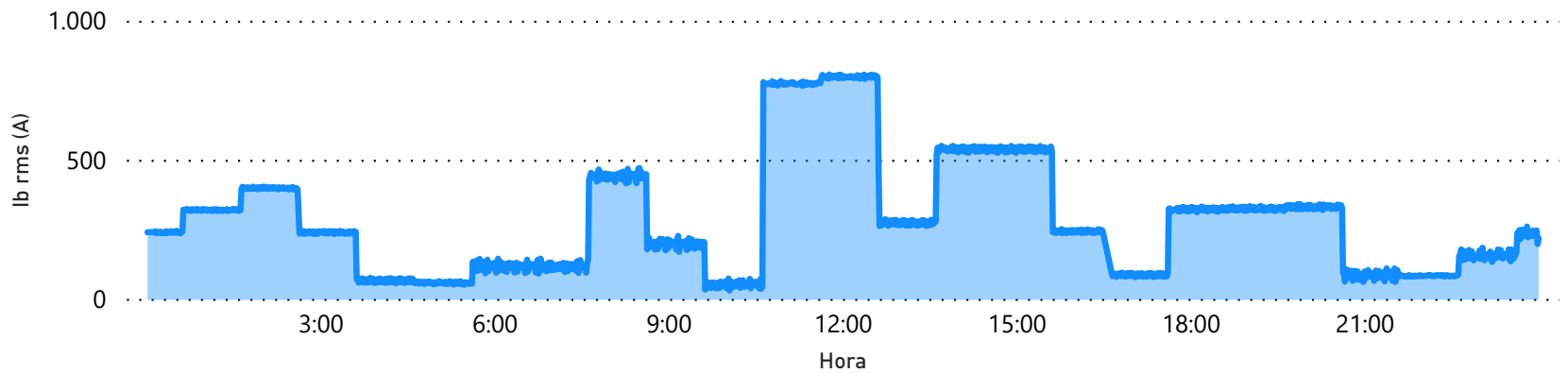
| Máx. de Ib_LV | Mín. de Ib_LV |
|---------------|---------------|
| 810,70 | 33,40 |

| Máx. de Ic_LV | Mín. de Ic_LV |
|---------------|---------------|
| 804,00 | 29,70 |

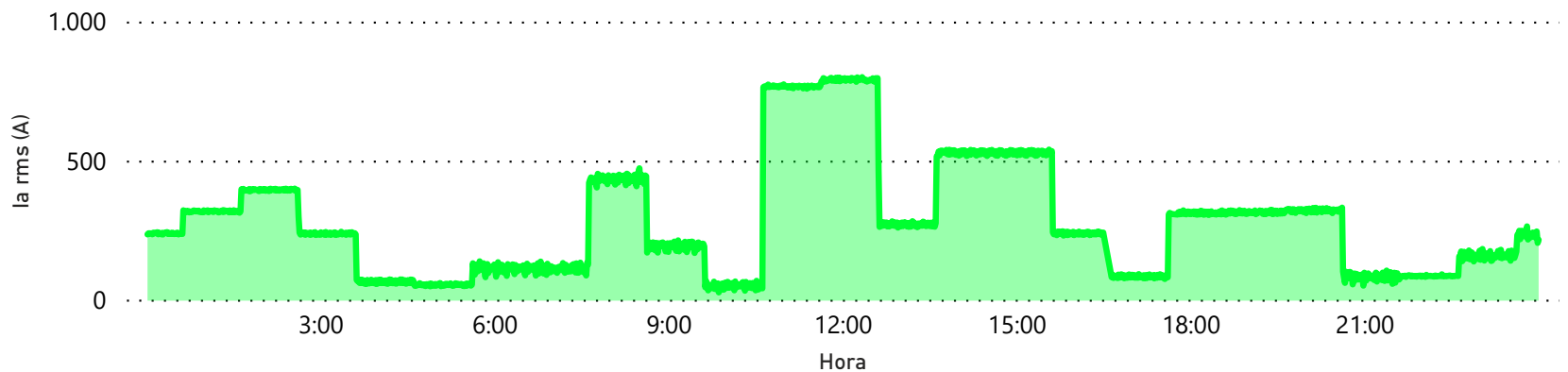
Intensidad eficaz fase a



Intensidad eficaz fase b



Intensidad eficaz fase c





INFORME DIARIO DE OPERACIÓN DEL BESS

NRG.Lab

ÍNDICE

| | |
|-----------------------|---|
| INTRODUCCIÓN | 2 |
| ESTUDIO DE POTENCIAS | 3 |
| ESTUDIO DE TENSIONES | 4 |
| ESTUDIO DE CORRIENTES | 5 |

INTRODUCCIÓN

El objetivo de este documento es mostrar un informe sobre el funcionamiento del BESS a lo largo de un día.

Se plantea un estudio y análisis de las diferentes señales de interés obtenidas, para así poder visualizar la operación del sistema i poder detectar cualquier comportamiento anómalo.

Para autocompletar el documento, es necesario indicar qué fecha se desea visualizar en el informe.

jueves, 4 de junio de 2020

Selección Fecha para Potencias

2020 (Año) + junio (Mes) ... ✓

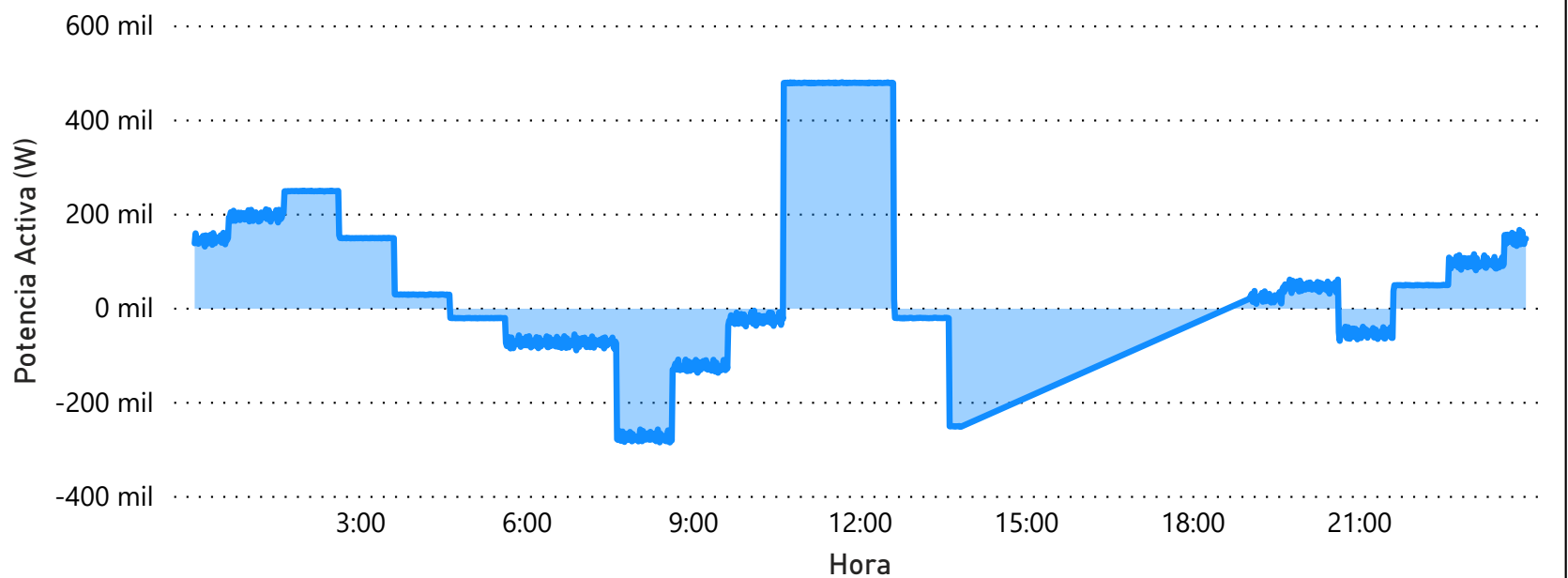
Selección Fecha para Tensiones y Corrientes

2020 (Año) + junio (Mes) ... ✓

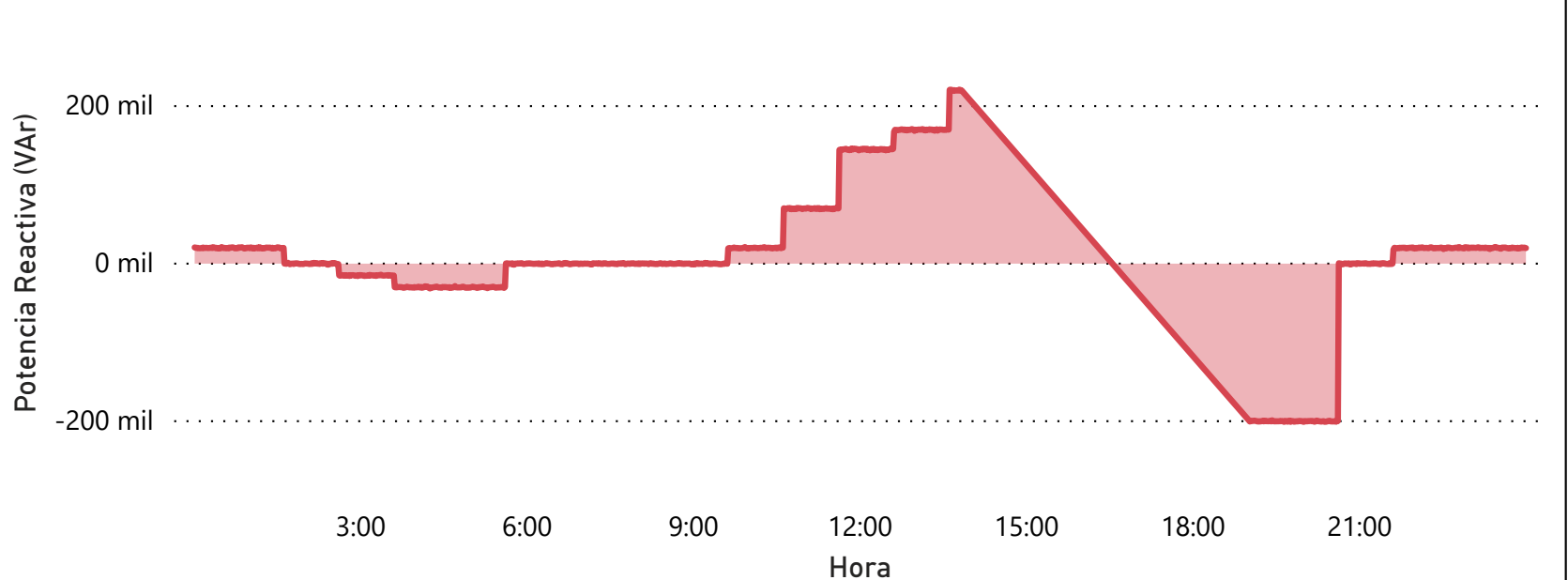
ESTUDIO DE POTENCIAS

Los datos obtenidos durante el jueves, 4 de junio de 2020 indican que los valores de potencia máximos en la salida del BESS han sido de 481,42 mil W y 220,88 mil VAR, y los valores mínimos han sido de -284,32 mil W y -200,86 mil VAR, respectivamente. Siendo los valores promedios diarios de 64,46 mil W y 7,39 mil VAR.

Potencia Activa



Potencia Reactiva



ESTUDIO DE TENSIONES

Los datos obtenidos durante el **jueves, 4 de junio de 2020** indican que los valores promedio de tensión en la salida del convertidor, en Voltios, han sido:

Promedio de Va_LV



207,48

Promedio de Vb_LV



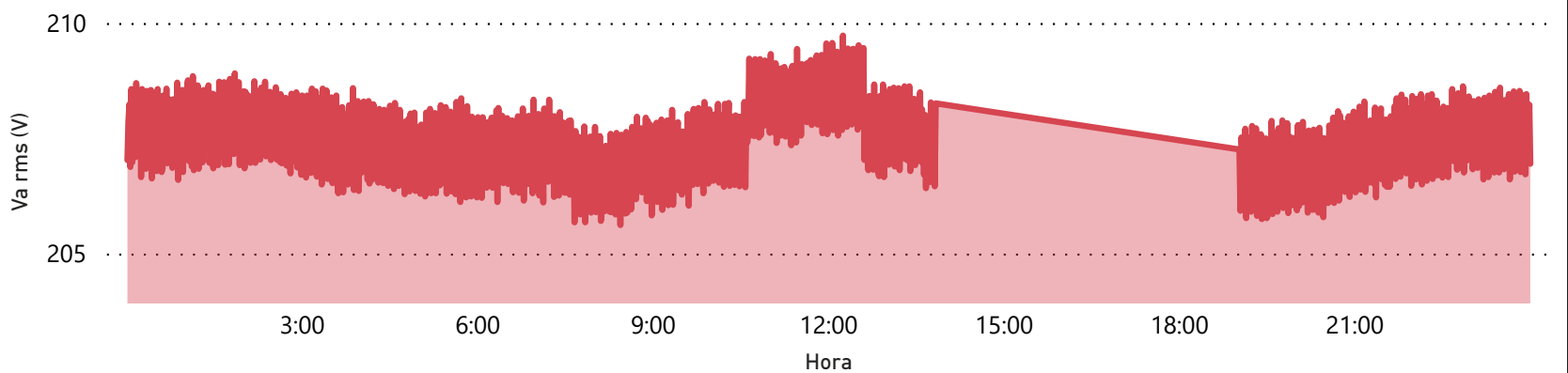
207,57

Promedio de Vc_LV

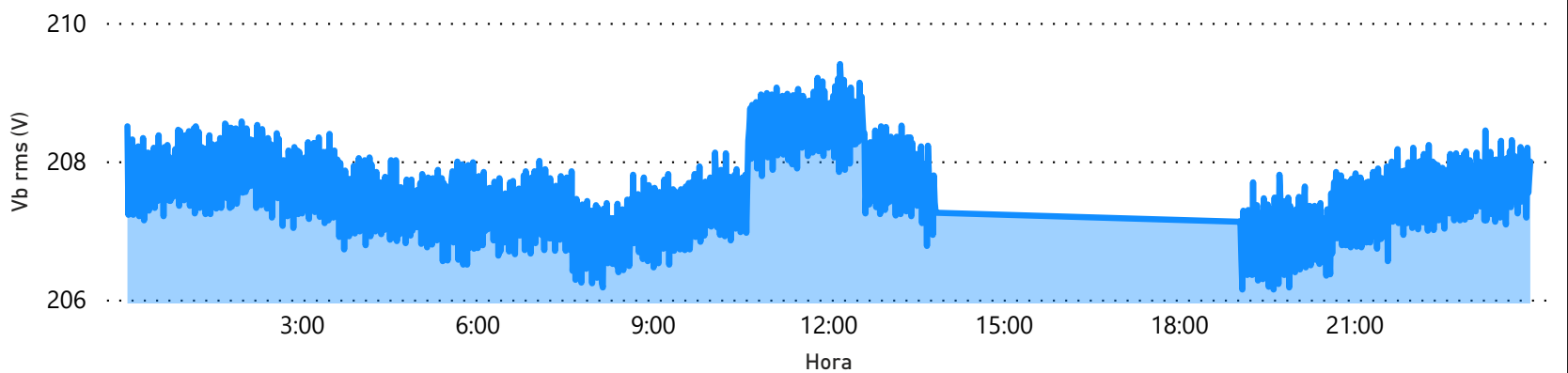


208,55

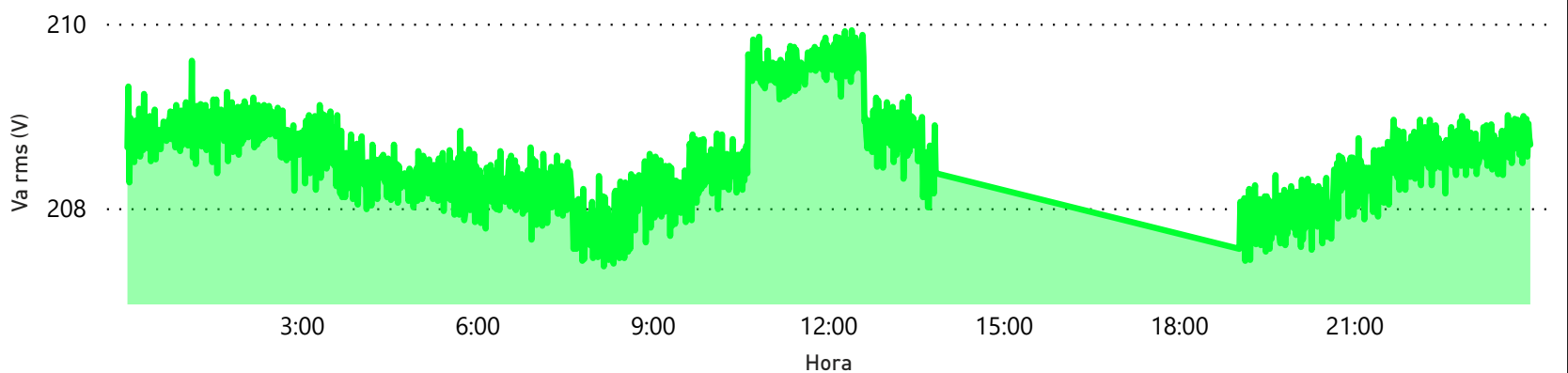
Tensión eficaz fase a



Tensión eficaz fase b



Tensión eficaz fase c

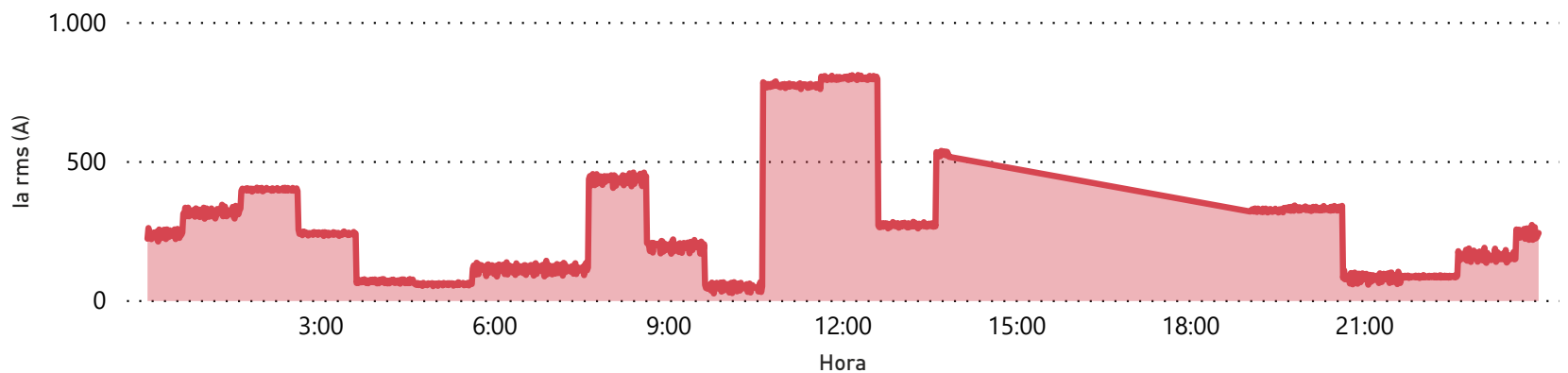


ESTUDIO DE CORRIENTES

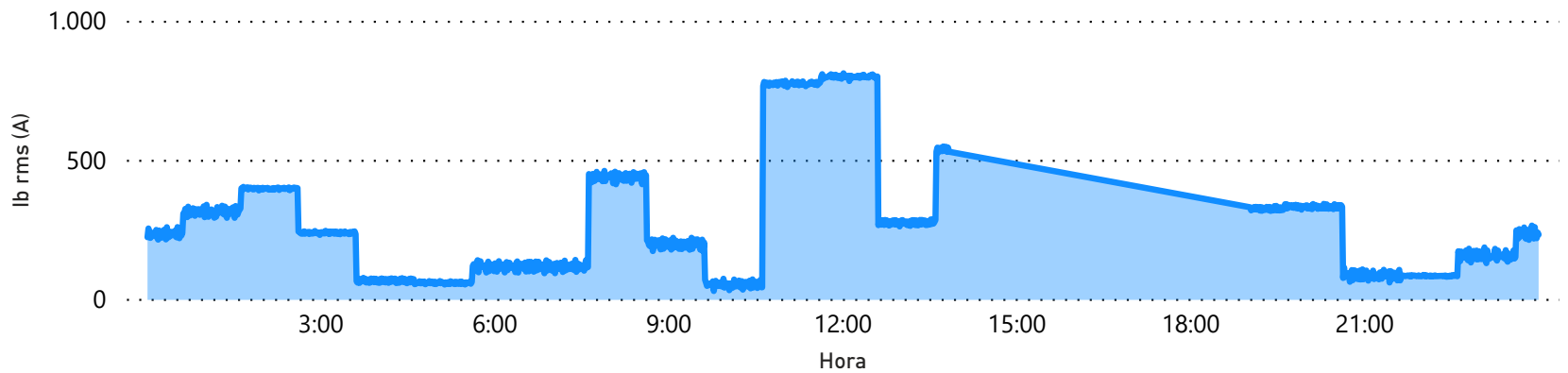
Los datos obtenidos durante el **jueves, 4 de junio de 2020** indican que los valores de las corrientes, en Amperios, han sido:

| Máx. de Ia_LV | Mín. de Ia_LV | Máx. de Ib_LV | Mín. de Ib_LV | Máx. de Ic_LV | Mín. de Ic_LV |
|---------------|---------------|---------------|---------------|---------------|---------------|
| 813,00 | 27,20 | 814,80 | 31,90 | 806,90 | 32,00 |

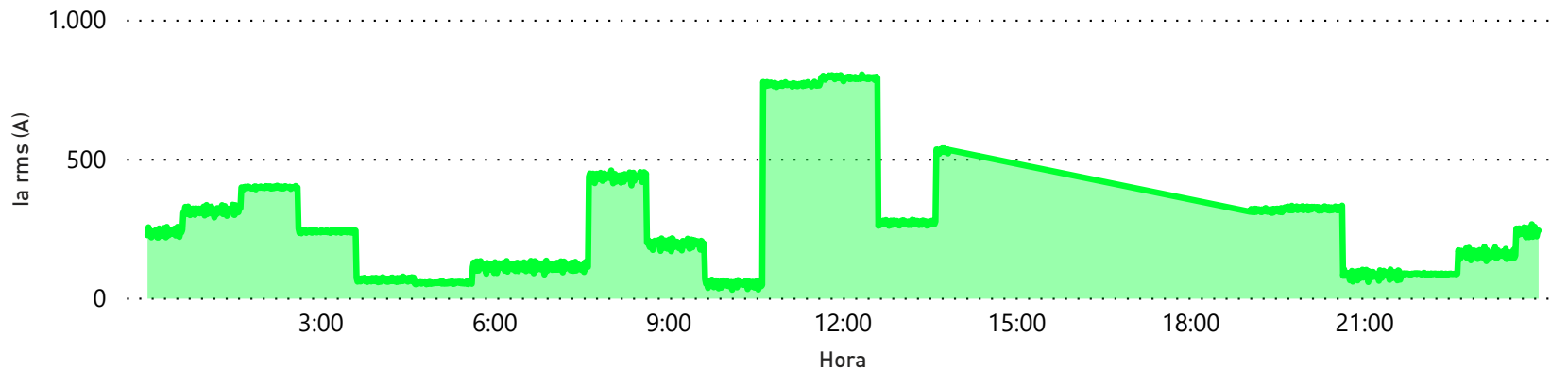
Intensidad eficaz fase a



Intensidad eficaz fase b



Intensidad eficaz fase c



ANNEX 5

CODI DE PROGRAMACIÓ DEL SCADA

```

//#####//
// This file contains functions used call different menú screens
//#####//

//Go to menú Screen

FUNCTION callHome()

    PageDisplay("Home");

END

//Go to Operation screen only if there is a user logged in, if not, go to Home screen.

FUNCTION callOperation()

    IF UserInfo(0)<>"0" THEN
        PageDisplay("Operation");
    ELSE
        PageDisplay("Home");
        PLSLoginUser_NRGLab();
    END
END

//Go to Scheduling screen only if there is a user logged in, if not, go to Home screen.

FUNCTION callScheduling()

    IF UserInfo(0)<>"0" THEN
        PageDisplay("Scheduling");
    ELSE
        PageDisplay("Home");
        PLSLoginUser_NRGLab();
    END
END

//Go to Monitoring screen only if there is a user logged in, if not, go to Home screen.

FUNCTION callAnalysis()

    IF UserInfo(0)<>"0" THEN
        PageDisplay("Monitoring");
        callMonitoring();    !When enter to Monitoring screen, upload graphic document
    ELSE
        PageDisplay("Home");
        PLSLoginUser_NRGLab();
    END
END

//Go to Parametrization MB screen only if there is a user logged in, if not, go to Home screen.

FUNCTION callFLCMB()

    IF UserInfo(0)<>"0" THEN
        PageDisplay("Parametrization FLC MB");
    ELSE
        PageDisplay("Home");
        PLSLoginUser_NRGLab();
    END
END

//Go to Parametrization EAP screen only if there is a user logged in, if not, go to Home screen.

FUNCTION callFLCEAP()

    IF UserInfo(0)<>"0" THEN
        PageDisplay("Parametrization FLC EAP");
    ELSE
        PageDisplay("Home");
        PLSLoginUser_NRGLab();
    END
END

```

```
//Go to Parametrization SLC screen only if there is a user logged in, if not, go to Home screen.
```

```
FUNCTION callSLC()
```

```
    IF UserInfo(0)<>"0" THEN  
        PageDisplay("Parametrization SLC");  
    ELSE  
        PageDisplay("Home");  
        PLSLoginUser_NRGLab();  
    END
```

```
END
```

```
//Go to Alarms screen only if there is a user logged in, if not, go to Home screen.
```

```
FUNCTION callAlarms()
```

```
    IF UserInfo(0)<>"0" THEN  
        PageDisplay("Alarms");  
        callMonitoring();  
    ELSE  
        PageDisplay("Home");  
        PLSLoginUser_NRGLab();  
    END
```

```
END
```

```

//#####//

//This file contains the functions used in the Operation Screen

//#####//

GLOBAL INT EmergencyPCS=0; !Variable for intermittance in Emergency State

//#####//

//Function to change value of enable bit of service.
//Service is selected by tag and i values, where 'i' is the tag number (according to the list of variables in
ascendent scale)

FUNCTION Op_Schedule(INT value, INT i, INT tag)

    INT n;

    aDigitalValue[i][0]=value;

    n=Int_Conversion(i);

    tag=n;

END

//This function returns an hour value depending on the position given when it is called.
//Hours from 0 to 23.
//Used for rolling schedule.

INT FUNCTION Op_ShowRollingHour(INT i)

    INT value;
    INT hour=TimeHour(TimeCurrent());

    value=hour+i;

    IF value>=24 THEN
        value=value-24;
    END

    RETURN value;

END

//Show FLC service state in function of current comms protocol

INT FUNCTION Op_ShowServiceStateFLC(INT i, INT j)

    INT value;

    IF FLC_Comms=1 THEN                !IF MB active
        value=aDigitalValue[i][0];    !i is pointer for MB command
    ELSE                                !If EAP active
        value=aDigitalValue[j][0];    !i is pointer for EAP command
    END

    RETURN value;

END

//Function to show the value of the Active Power reference depending the hour.
//If the local variable which denotes if the system is in Manual or Programmed mode is =1 (Programmed), the array
variable of the Pred will be shown, if not, the tag value of Pref

REAL FUNCTION Op_ShowPref(INT i)

    REAL value;

    IF Op_ProgMan_Mode=1 THEN

        value=Op_Pref[Op_ShowRollingHour(i)];

    ELSE

        value=NRGLAB_Device\FLCin_input_ActivePowerRef

    END

END

```

```

END

RETURN value;

END

//Function to show the value of the Reactive Power refence depending the hour.
//If the local variable which denotates if the system is in Manual or Programed mode is =1 (Porgramed), the array
variable of the Pred will be shown, if not, the tag value of Qref

REAL FUNCTION Op_ShowQref(INT i)

    REAL value;

    IF Op_ProgMan_Mode=1 THEN

        value=Op_Qref[Op_ShowRollingHour(i)];

    ELSE

        value=NRGLAB_Device\FLCin_input_ReactivePowerRef

    END

    RETURN value;

END

//Change value of the Pref[i] where the position is found by the relation with the current hour and the box
position.
//Value is changed via a pop-up box called by Input function

FUNCTION Op_ChangePref(INT i)

    REAL value;
    STRING new;
    STRING ongoing;

    INT j=Op_ShowRollingHour(i); ! Get value of array position related to hour

    ongoing=RealToStr(Op_Pref[j],8,2,".");    ! Convert Real value to String to have appropriate format for Input
function

    new=Input("Active Power Reference", "Value", ongoing);    ! Input function where a pop-up box opens and the
current value is displayed and can be changed

    IF IsError()=299 THEN    ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

    value=StrToReal(new);    ! Convert String value result to Real again

    Op_Pref[j]=value;

END

//Change value of the Qref[i] where the position is found by the relation with the current hour and the box
position.
//Value is changed via a pop-up box called by Input function

FUNCTION Op_ChangeQref(INT i)

    REAL value;
    STRING new;
    STRING ongoing;

    INT j=Op_ShowRollingHour(i); ! Get value of array position related to hour

    ongoing=RealToStr(Op_Qref[j],8,2,".");    ! Convert Real value to String to have appropriate format for Input
function

    new=Input("Reactive Power Reference", "Value", ongoing);    ! Input function where a pop-up box opens and the
current value is displayed and can be changed

    IF IsError()=299 THEN    ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

END

```

```

    value=StrToReal(new);    ! Convert String value result to Real again

    Op_Qref[j]=value;

END

GLOBAL INT aDIntValue[32];

//Function called when Programed or Manual mode are choosed
//As their tags is a Double Int and Porgramed corresponds to bit n° 11 and Manual to bit n° 12, the tag have to be
passed in Digital form

FUNCTION Op_ChooseProgManMode(INT i)

    INT j=0;

    REAL m;

    INT n=NRGLAB_Device\TLCInput_Command;    ! Pass tag value to a variable

    WHILE j<16 DO    ! Convert DInt value to Digital

        aDIntValue[j]=n MOD 2;
        n=n/2;
        j=j+1;

    END

    IF i=1 THEN    ! If Programed button have been pressed, bit n° 11 =1, bit n° 10=0 and local
project variable which denotated mode state turn to 1

        //aDIntValue[10]=1;
        aDIntValue[11]=1;    //previously was 0
        Op_ProgMan_Mode=1;

    END

    IF i=0 THEN    ! If Manual button have been pressed, bit n° 11=0, bit n° 10=1 and local project
variable which denotated mode state turn to 0

        //aDIntValue[10]=0;
        aDIntValue[11]=1;
        Op_ProgMan_Mode=0;

    END

    j=0;

    WHILE j<32 DO    ! Convert Digital Value to DInt value again

        m = m + aDIntValue[j]*(Pow(2,j));
        j=j+1;

    END

    NRGLAB_Device\TLCInput_Command=m;    ! Transfer the value obtained to the tag.

END

//Function to open the pop-up box to change the value of the Active or Reactive Power Reference

REAL FUNCTION Change_ManRef(REAL tag, STRING title)

    STRING new;
    STRING ongoing=RealToStr(tag,8,2,".");    ! Convert Real value to String to have appropriate format for Input
function

    new=Input("Manual Reference", title, ongoing);    ! Input function where a pop-up box opens and the
current value is displayed and can be changed

    IF IsError()=299 THEN    ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

    RETURN StrToReal(new);    ! Convert String value result to Real again

END

```

```
//Intermittance for PCS State when is on Emergency State
```

```
INT FUNCTION Special_EmergencyPCS(INT i)
```

```
    INT j;
```

```
    IF NRGLAB_Device\FLCin_statVar_Status=100 THEN
```

```
        WHILE j<i DO
```

```
            j=j+1;
```

```
        END
```

```
        Toggle(EmergencyPCS);
```

```
    ELSE
```

```
        EmergencyPCS=0;
```

```
    END
```

```
    RETURN EmergencyPCS;
```

```
END
```

```

//#####//

//This file contains functions called and operated into Scheduling Screen

//#####//

//State service bits to know if all scheduled hours are ON

GLOBAL INT bSchIE;
GLOBAL INT bSchFR;
GLOBAL INT bSchMF;
GLOBAL INT bSchVR;
GLOBAL INT bSchRVM;
GLOBAL INT bSchPSM;
GLOBAL INT bSchRRE;
GLOBAL INT bSchPCM;
GLOBAL INT bSchPBP;

//#####//

//When the Enable/Disabte all service button is clicked, this function is called
//It uses an integer as a pointer to know which service is.
//In each case (service), it is asked whereas if bit scheduling service state is true (1) or false (0),
//by this way, THEN all the array service positions are set TO 0 OR 1 (enabled OR disabled).

FUNCTION Sch_SetAllService(INT i)

    INT j;

    SELECT CASE i

        CASE 1 ! Inertia Emulation

            IF bSchIE=0 THEN
                FOR j=0 TO 23 DO
                    aSch_IE[j]=1;
                    bSch_Changed_IE[j]=1;
                END
                bSchIE=1;
            ELSE
                FOR j=0 TO 23 DO
                    aSch_IE[j]=0;
                    bSch_Changed_IE[j]=1;
                END
                bSchIE=0;
            END

        CASE 2 ! Frequency Regulation

            IF bSchFR=0 THEN
                FOR j=0 TO 23 DO
                    aSch_FR[j]=1;
                    bSch_Changed_FR[j]=1;
                END
                bSchFR=1;
            ELSE
                FOR j=0 TO 23 DO
                    aSch_FR[j]=0;
                    bSch_Changed_FR[j]=1;
                END
                bSchFR=0;
            END

        CASE 3 ! ModalFrequency

            IF bSchMF=0 THEN
                FOR j=0 TO 23 DO
                    aSch_MF[j]=1;
                    bSch_Changed_MF[j]=1;
                END
                bSchMF=1;
            ELSE
                FOR j=0 TO 23 DO
                    aSch_MF[j]=0;
                    bSch_Changed_MF[j]=1;
                END
                bSchMF=0;
            END

    END

```



```

CASE 4 ! Voltage Regulation

    IF bSchVR=0 THEN
        FOR j=0 TO 23 DO
            aSch_VR[j]=1;
            bSch_Changed_VR[j]=1;
        END
        bSchVR=1;
    ELSE
        FOR j=0 TO 23 DO
            aSch_VR[j]=0;
            bSch_Changed_VR[j]=1;
        END
        bSchVR=0;
    END

CASE 5 ! Resource Variability

    IF bSchRVM=0 THEN
        FOR j=0 TO 23 DO
            aSch_RVM[j]=1;
            bSch_Changed_RVM[j]=1;
        END
        bSchRVM=1;
    ELSE
        FOR j=0 TO 23 DO
            aSch_RVM[j]=0;
            bSch_Changed_RVM[j]=1;
        END
        bSchRVM=0;
    END

CASE 6 ! Power Shifting

    IF bSchPSM=0 THEN
        FOR j=0 TO 23 DO
            aSch_PSM[j]=1;
            bSch_Changed_PSM[j]=1;
        END
        bSchPSM=1;
    ELSE
        FOR j=0 TO 23 DO
            aSch_PSM[j]=0;
            bSch_Changed_PSM[j]=1;
        END
        bSchPSM=0;
    END

CASE 7 ! Ramp Rate Enhancement

    IF bSchRRE=0 THEN
        FOR j=0 TO 23 DO
            aSch_RRE[j]=1;
            bSch_Changed_RRE[j]=1;
        END
        bSchRRE=1;
    ELSE
        FOR j=0 TO 23 DO
            aSch_RRE[j]=0;
            bSch_Changed_RRE[j]=1;
        END
        bSchRRE=0;
    END

CASE 8 ! Power Curtailment

    IF bSchPCM=0 THEN
        FOR j=0 TO 23 DO
            aSch_PCM[j]=1;
            bSch_Changed_PCM[j]=1;
        END
        bSchPCM=1;
    ELSE
        FOR j=0 TO 23 DO
            aSch_PCM[j]=0;
            bSch_Changed_PCM[j]=1;
        END
        bSchPCM=0;
    END

CASE 9 ! Power Balance

```

```

        IF bSchPBP=0 THEN
            FOR j=0 TO 23 DO
                aSch_PBP[j]=1;
                bSch_Changed_PBP[j]=1;
            END
            bSchPBP=1;
        ELSE
            FOR j=0 TO 23 DO
                aSch_PBP[j]=0;
                bSch_Changed_PBP[j]=1;
            END
            bSchPBP=0;
        END
    END SELECT

END

```

//In this function it is checked the whole service state.
 //By looking over all the service array it can be known if all the positions (hours) have the same value or not

```

FUNCTION Sch_SetStateBitService(INT i)

```

```

    INT check;
    INT j;

```

```

    SELECT CASE i

```

```

        CASE 1 !Inertia Emualtion

```

```

            check=1;

            FOR j=0 TO 23 DO
                IF aSch_IE[j]=0 THEN
                    check=0;
                END
            END

            IF check=1 THEN
                bSchPBP=1;
            ELSE
                bSchPBP=0;
            END

```

```

        CASE 2 !Frequency Regulation

```

```

            check=1;

            FOR j=0 TO 23 DO
                IF aSch_FR[j]=0 THEN
                    check=0;
                END
            END

            IF check=1 THEN
                bSchFR=1;
            ELSE
                bSchFR=0;
            END

```

```

        CASE 3 !Modal Frequency

```

```

            check=1;

            FOR j=0 TO 23 DO
                IF aSch_MF[j]=0 THEN
                    check=0;
                END
            END

            IF check=1 THEN
                bSchMF=1;
            ELSE
                bSchMF=0;
            END

```

```

        CASE 4 !Voltage Regulation

```

```

            check=1;

            FOR j=0 TO 23 DO

```

```

        IF aSch_VR[j]=0 THEN
            check=0;
        END
    END

    IF check=1 THEN
        bSchVR=1;
    ELSE
        bSchVR=0;
    END

CASE 5 !Resource Variability

    check=1;

    FOR j=0 TO 23 DO
        IF aSch_RVM[j]=0 THEN
            check=0;
        END
    END

    IF check=1 THEN
        bSchRVM=1;
    ELSE
        bSchRVM=0;
    END

CASE 6 !Power Shifting

    check=1;

    FOR j=0 TO 23 DO
        IF aSch_PSM[j]=0 THEN
            check=0;
        END
    END

    IF check=1 THEN
        bSchPSM=1;
    ELSE
        bSchPSM=0;
    END

CASE 7 !Ramp Rate Enhancement

    check=1;

    FOR j=0 TO 23 DO
        IF aSch_RRE[j]=0 THEN
            check=0;
        END
    END

    IF check=1 THEN
        bSchRRE=1;
    ELSE
        bSchRRE=0;
    END

CASE 8 !Power Curtailment

    check=1;

    FOR j=0 TO 23 DO
        IF aSch_PCM[j]=0 THEN
            check=0;
        END
    END

    IF check=1 THEN
        bSchPCM=1;
    ELSE
        bSchPCM=0;
    END

CASE 9 !Power Balance

    check=1;

    FOR j=0 TO 23 DO
        IF aSch_PBP[j]=0 THEN
            check=0;
        END
    END

```

```

        END

        IF check=1 THEN
            bSchPBP=1;
        ELSE
            bSchPBP=0;
        END

    END SELECT

END

END

//This function permits to show the service state bit.

INT FUNCTION Sch_ShowStateBitService(INT i)

    INT value;

    Sch_SetStateBitService(i);

    SELECT CASE i

        CASE 1
            value=bSchIE;
        CASE 2
            value=bSchFR;
        CASE 3
            value=bSchMF;
        CASE 4
            value=bSchVR;
        CASE 5
            value=bSchRVM;
        CASE 6
            value=bSchPSM;
        CASE 7
            value=bSchRRE;
        CASE 8
            value=bSchPCM;
        CASE 9
            value=bSchPBP;
    END SELECT

    RETURN value;

END

//Save changes done in Schedule
//Changed services updated and state bit turns to 0.

FUNCTION Sch_Save_ServiceSchedule()

    INT i;

    FOR i=0 TO 23 DO

        aSch_Saved_IE[i]=aSch_IE[i];
        bSch_Changed_IE[i]=0;

        aSch_Saved_FR[i]=aSch_FR[i];
        bSch_Changed_FR[i]=0;

        aSch_Saved_MF[i]=aSch_MF[i];
        bSch_Changed_MF[i]=0;

        aSch_Saved_VR[i]=aSch_VR[i];
        bSch_Changed_VR[i]=0;

        aSch_Saved_RVM[i]=aSch_RVM[i];
        bSch_Changed_RVM[i]=0;

        aSch_Saved_PSM[i]=aSch_PSM[i];
        bSch_Changed_PSM[i]=0;

        aSch_Saved_RRE[i]=aSch_RRE[i];
        bSch_Changed_RRE[i]=0;

        aSch_Saved_PCM[i]=aSch_PCM[i];
        bSch_Changed_PCM[i]=0;

        aSch_Saved_PBP[i]=aSch_PBP[i];
        bSch_Changed_PBP[i]=0;
    
```

```

END

END

//Saving as Administrator, giving privilege by enabling admin bit to those
//services changed by admin.

FUNCTION Sch_Save_AsAdministrator_ServiceSchedule()

    INT i;

    FOR i=0 TO 23 DO

        IF bSch_Changed_IE[i]=1 THEN
            bSch_Admin_IE[i]=1;
            bSchAdminIE=1;
        END

        IF bSch_Changed_FR[i]=1 THEN
            bSch_Admin_FR[i]=1;
            bSchAdminFR=1;
        END

        IF bSch_Changed_MF[i]=1 THEN
            bSch_Admin_MF[i]=1;
            bSchAdminMF=1;
        END

        IF bSch_Changed_VR[i]=1 THEN
            bSch_Admin_VR[i]=1;
            bSchAdminVR=1;
        END

        IF bSch_Changed_RVM[i]=1 THEN
            bSch_Admin_RVM[i]=1;
            bSchAdminRVM=1;
        END

        IF bSch_Changed_PSM[i]=1 THEN
            bSch_Admin_PSM[i]=1;
            bSchAdminPSM=1;
        END

        IF bSch_Changed_RRE[i]=1 THEN
            bSch_Admin_RRE[i]=1;
            bSchAdminRRE=1;
        END

        IF bSch_Changed_PCM[i]=1 THEN
            bSch_Admin_PCM[i]=1;
            bSchAdminPCM=1;
        END

        IF bSch_Changed_PBP[i]=1 THEN
            bSch_Admin_PBP[i]=1;
            bSchAdminPBP=1;
        END

    END

    Sch_Save_ServiceSchedule();

END

//When saving but not as Administrator al admin bits are disabled then if
//current user is an administrator

FUNCTION Sch_Save_NotAsAdmin()

    INT i;

    IF GetPriv(6,0) THEN

        FOR i=0 TO 23 DO

            bSch_Admin_IE[i]=0;
            bSch_Admin_FR[i]=0;
            bSch_Admin_MF[i]=0;
            bSch_Admin_VR[i]=0;
            bSch_Admin_RVM[i]=0;

```

```

        bSch_Admin_PSM[i]=0;
        bSch_Admin_RRE[i]=0;
        bSch_Admin_PCM[i]=0;
        bSch_Admin_PBP[i]=0;

```

```

END

```

```

bSchAdminIE=0;
bSchAdminFR=0;
bSchAdminMF=0;
bSchAdminVR=0;
bSchAdminRVM=0;
bSchAdminPSM=0;
bSchAdminRRE=0;
bSchAdminPCM=0;
bSchAdminPBP=0;

```

```

END

```

```

END

```

```

//Undo changes not saved

```

```

FUNCTION Sch_UndoService()

```

```

    INT i;

```

```

    FOR i=0 TO 23 DO

```

```

        aSch_IE[i]=aSch_Saved_IE[i];
        bSch_Changed_IE[i]=0;

```

```

        aSch_FR[i]=aSch_Saved_FR[i];
        bSch_Changed_FR[i]=0;

```

```

        aSch_MF[i]=aSch_Saved_MF[i];
        bSch_Changed_MF[i]=0;

```

```

        aSch_VR[i]=aSch_Saved_VR[i];
        bSch_Changed_VR[i]=0;

```

```

        aSch_RVM[i]=aSch_Saved_RVM[i];
        bSch_Changed_RVM[i]=0;

```

```

        aSch_PSM[i]=aSch_Saved_PSM[i];
        bSch_Changed_PSM[i]=0;

```

```

        aSch_RRE[i]=aSch_Saved_RRE[i];
        bSch_Changed_RRE[i]=0;

```

```

        aSch_PCM[i]=aSch_Saved_PCM[i];
        bSch_Changed_PCM[i]=0;

```

```

        aSch_PBP[i]=aSch_Saved_PBP[i];
        bSch_Changed_PBP[i]=0;

```

```

    END

```

```

END

```

```

//Know if current user has Admin privilege or not

```

```

INT FUNCTION Sch_NotPrivilege(INT bAdmin)

```

```

    INT value=0;

```

```

    IF GetPriv(6,0)=0 AND bAdmin=1 THEN
        value=1;

```

```

    END

```

```

    RETURN value;

```

```

END

```

```

//Function to change Active Power reference value in the schedule

```

```

FUNCTION Sch_ChangePref(INT i)

```

```

    REAL value;
    STRING new;

```

```

    STRING ongoing;

    ongoing=RealToStr(Sch_Pref[i],8,2,"."); ! Convert Real value to String to have appropriate format for Input
function

    new=Input("Active Power Reference", "Value", ongoing); ! Input function where a pop-up box opens and the
current value is displayed and can be changed

    IF IsError()=299 THEN ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

    value=StrToReal(new); ! Convert String value result to Real again

    Sch_Pref[i]=value

END

//Function to change Reactive Power reference value in the schedule
FUNCTION Sch_ChangeQref(INT i)

    REAL value;
    STRING new;
    STRING ongoing;

    ongoing=RealToStr(Sch_Pref[i],8,2,"."); ! Convert Real value to String to have appropriate format for Input
function

    new=Input("Reactive Power Reference", "Value", ongoing); ! Input function where a pop-up box opens and the
current value is displayed and can be changed

    IF IsError()=299 THEN ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

    value=StrToReal(new); ! Convert String value result to Real again

    Sch_Qref[i]=value

END

//Function to change Power Reserve value from a Service
FUNCTION Sch_ChangePowerReserve(REAL tag, STRING title)

    REAL value;
    STRING new;
    STRING ongoing;

    ongoing=RealToStr(tag,8,2,"."); ! Convert Real value to String to have appropriate format for Input function

    new=Input(title, "Value", ongoing); ! Input function where a pop-up box opens and the current value is
displayed and can be changed

    IF IsError()=299 THEN ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

    value=StrToReal(new); ! Convert String value result to Real again

    tag=value

END

```

```

//#####//
// This file contains functions used in Monitoring screen
//#####//

//Upload graphic document to be shown in the Process Analyst

FUNCTION callMonitoring()

    SleepMS(250);

    OBJECT plot1 = ObjectByName("AN62");
    _ObjectCallMethod(plot1, "LoadFromFile", "Monitoring V and I converter.pav", 0);

END

//PCS Status is Long type variable sending PCS State as a number
//In this function, according to the value returns an string with state value
//By this way, it is visualized as a text the current PCS Status

STRING FUNCTION PCS_State(INT i)

    STRING value;

    SELECT CASE i

        CASE 0
            value="Off State";    !If PCS Status = 0, return "Off State"
        CASE 1
            value="Idle State";    !If PCS Status = 0, return "Idle State"
        CASE 2
            value="Manual Mode State";    !If PCS Status = 0, return "Manual Mode State"
        CASE 3
            value="Average Power State"; !If PCS Status = 0, return "Average Power State"
        CASE 4
            value="Instant Power State"; !If PCS Status = 0, return "Instant Power State"
        CASE 5
            value="Ramp Down State"; !If PCS Status = 0, return "Ramp Down State"
        CASE 100
            value="Emergency State"; !If PCS Status = 0, return "Emergency State"
    END SELECT

    RETURN value;

END

//To show FLC Modbus or EAP service state according to the current State Machine Status

INT FUNCTION Mon_ServiceState(INT i, INT j)

    INT value;

    IF NRGLAB_Device\FLCin_statVar_Status=4 THEN !If Instant Pow State show EAP service
        value=aDigitalValue[j][0];
    ELSE
        !Else show MB service
        value=aDigitalValue[i][0];
    END

    RETURN value;

END

```



```

//#####//
/*Functions used in Parametrization Screens for Parameter Values      */
//#####//

//Reset to all values (Param, Command, Max, Min and Default).
//Showing again previous values before being changed AND NOT saved

FUNCTION Reset_Service(INT i, INT j, INT k)

    Reset_ServiceParameter(i,j);
    Reset_ServiceMaximum(i,j);
    Reset_ServiceMinimum(i,j);
    Reset_ServiceDefault(i,j);
    Reset_ServiceComand(k);

END

//Change value from Parameter. Must be inside bound of Max and Min

FUNCTION Change_Parameter(INT i, STRING title)

    REAL value;
    STRING new;
    STRING ongoing=RealToStr(aParam[i],8,3,"."); ! Convert Real value to String to have appropriate format for Input
function

    new=Input(title, "Value", ongoing);           ! Input function where a pop-up box opens and the current value is
displayed and can be changed

    IF IsError()=299 THEN      ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

    value=StrToReal(new);      ! Convert String value result to Real again

    IF value <> aParam[i] THEN  ! Only if value have changed

        IF value>aMax_initial[i] OR value<aMin_initial[i] THEN
            Message("Error","Entered value is out of bounds",48);
        ELSE
            Change_StateParameter(i);
            aParam[i]=value;
        END

    END

END

END

//Get Default value as Parameter value

FUNCTION GetDeafult_Parameter(INT i)

    Change_StateParameter(i);
    aParam[i]=aDef_initial[i];

END

//Change value to '1' from state Parameter bit for i position

FUNCTION Change_StateParameter(INT i)

    bState_Param[i]=1;

END

//Change value to '0' from state Parameter bit for all positions

FUNCTION Reset_StateParameter()

    INT i;

    FOR i=0 TO 323 DO
        bState_Param[i]=0;
    END

```

```

END

//Reset Service by turning all bState_Param[i] of service to 0
FUNCTION Reset_ServiceParameter(INT i, INT j)
    FOR i=i TO j DO
        bState_Param[i]=0;
    END
END

//Tags get value from Parameter array. State Parameter bit for each position is set to '0'
FUNCTION Update_Parameter()
    ParamToTagUpdate();
    SleepMS(50);    ! Waiting 50ms in order to ensure that the updating operation is done properly
    Reset_StateParameter();
END

//State Parameter bit for each position is set to '0'
FUNCTION Undo_Parameter()
    Reset_StateParameter();
END

//Passing Parameters values to PLC tags
FUNCTION ParamToTagUpdate()
    IF bState_Param[20]=1 THEN
        NRGLAB_Device\FLCparamMB_COMM_NominalPower=aParam[20]*1000;
    END
    IF bState_Param[21]=1 THEN
        NRGLAB_Device\FLCparamMB_COMM_NominalVoltage=aParam[21];
    END
    IF bState_Param[22]=1 THEN
        NRGLAB_Device\FLCparamMB_COMM_NominalFreq=aParam[22];
    END
    IF bState_Param[23]=1 THEN
        NRGLAB_Device\FLCparamMB_COMM_ActivePowerLimit=aParam[23]*1000;
    END
    IF bState_Param[24]=1 THEN
        NRGLAB_Device\FLCparamMB_COMM_ReactivePowerLim=aParam[24]*1000;
    END

    IF bState_Param[36]=1 THEN
        NRGLAB_Device\FLCparamMB_IE_Inertia_Constant=aParam[36];
    END
    IF bState_Param[37]=1 THEN
        NRGLAB_Device\FLCparamMB_IE_HighPassFiltConst=aParam[37];
    END
    IF bState_Param[38]=1 THEN
        NRGLAB_Device\FLCparamMB_IE_Max_ROCOF=aParam[38];
    END
    IF bState_Param[39]=1 THEN
        NRGLAB_Device\FLCparamMB_IE_PowerLimit_Inertia=aParam[39]*1000;
    END

    IF bState_Param[52]=1 THEN
        NRGLAB_Device\FLCparamMB_FR_DroopConstantP=aParam[52];
    END
    IF bState_Param[53]=1 THEN
        NRGLAB_Device\FLCparamMB_FR_Max_delta_Energy=aParam[53]*1000;
    END
    IF bState_Param[54]=1 THEN
        NRGLAB_Device\FLCparamMB_FR_PowerLimit_Freq=aParam[54]*1000;
    END
    IF bState_Param[55]=1 THEN

```

```

    NRGLAB_Device\FLCparamMB_FR_FrequencyCutOff=aParam[55];
END
IF bState_Param[56]=1 THEN
    NRGLAB_Device\FLCparamMB_FR_FreqDeadBand=aParam[56];
END

IF bState_Param[69]=1 THEN
    NRGLAB_Device\FLCparamMB_MFA_Gain=aParam[69];
END
IF bState_Param[70]=1 THEN
    NRGLAB_Device\FLCparamMB_MFA_BPF_CenterFreq=aParam[70];
END
IF bState_Param[71]=1 THEN
    NRGLAB_Device\FLCparamMB_MFA_PowerLimit_MF=aParam[71]*1000;
END

IF bState_Param[84]=1 THEN
    NRGLAB_Device\FLCparamMB_VR_DroopConstantV=aParam[84];
END
IF bState_Param[85]=1 THEN
    NRGLAB_Device\FLCparamMB_VR_PowerLimit_Voltage=aParam[85]*1000;
END
IF bState_Param[86]=1 THEN
    NRGLAB_Device\FLCparamMB_VR_VoltageCutOff=aParam[86];
END
IF bState_Param[87]=1 THEN
    NRGLAB_Device\FLCparamMB_VR_VoltDeadBand=aParam[87];
END

```

//-----//

//FLC: EAP Communication

//-----//

```

IF bState_Param[116]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_NominalPower=aParam[116]*1000;
END
IF bState_Param[117]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_NominalFreq=aParam[117];
END
IF bState_Param[118]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_NominalVolt=aParam[118];
END
IF bState_Param[119]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_ActivePowLim=aParam[119]*1000;
END
IF bState_Param[120]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_ReactivePowLim=aParam[120]*1000;
END
IF bState_Param[121]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_Delta_limits=aParam[121];
END
IF bState_Param[122]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_E_Limits=aParam[122];
END
IF bState_Param[123]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_Kp_ReactiveCtrl=aParam[123];
END
IF bState_Param[124]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_Ki_ReactiveCtrl=aParam[124];
END
IF bState_Param[125]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_VirtReactance=aParam[125];
END
IF bState_Param[126]=1 THEN
    NRGLAB_Device\FLCparamEAP_COMM_VirtResistor=aParam[126];
END

IF bState_Param[132]=1 THEN
    NRGLAB_Device\FLCparamEAP_IE_Inertia_Constant=aParam[132];
END
IF bState_Param[133]=1 THEN
    NRGLAB_Device\FLCparamEAP_IE_Damping_Constant=aParam[133];
END

```

```

IF bState_Param[134]=1 THEN
    NRGLAB_Device\FLCparamEAP_IE_HighPassFiltConst=aParam[134];
END
IF bState_Param[135]=1 THEN
    NRGLAB_Device\FLCparamEAP_IE_Max_ROCOF=aParam[135];
END

IF bState_Param[148]=1 THEN
    NRGLAB_Device\FLCparamEAP_FR_DroopConstantP=aParam[148];
END
IF bState_Param[149]=1 THEN
    NRGLAB_Device\FLCparamEAP_FR_Max_Delta_Energy=aParam[149]*1000;
END
IF bState_Param[150]=1 THEN
    NRGLAB_Device\FLCparamEAP_FR_PowerLimit_Freq=aParam[150]*1000;
END
IF bState_Param[151]=1 THEN
    NRGLAB_Device\FLCparamEAP_FR_LPF_CutOff_Freq=aParam[151];
END
IF bState_Param[152]=1 THEN
    NRGLAB_Device\FLCparamEAP_FR_DeadBand_Limits_P=aParam[152]*1000;
END

IF bState_Param[164]=1 THEN
    NRGLAB_Device\FLCparamEAP_MFA_Gain=aParam[164];
END
IF bState_Param[165]=1 THEN
    NRGLAB_Device\FLCparamEAP_MFA_PowerLimit=aParam[165]*1000;
END
IF bState_Param[166]=1 THEN
    NRGLAB_Device\FLCparamEAP_MFA_Max_ROCOF_Modal=aParam[166];
END
IF bState_Param[167]=1 THEN
    NRGLAB_Device\FLCparamEAP_MFA_BPF_CenterFreq=aParam[167];
END

IF bState_Param[180]=1 THEN
    NRGLAB_Device\FLCparamEAP_VR_DroopConstantQ=aParam[180];
END
IF bState_Param[181]=1 THEN
    NRGLAB_Device\FLCparamEAP_VR_LPF_CutOff_Volt=aParam[181];
END
IF bState_Param[182]=1 THEN
    NRGLAB_Device\FLCparamEAP_VR_DeadBand_Limit_Q=aParam[182];
END
IF bState_Param[183]=1 THEN
    NRGLAB_Device\FLCparamEAP_VR_PowerLimit_Volt=aParam[183]*1000;
END

```

```

//-----//
//SLC Communication
//-----//

```

```

IF bState_Param[212]=1 THEN
    NRGLAB_Device\SLCparam_COMM_BattSoC=aParam[212];
END
IF bState_Param[213]=1 THEN
    NRGLAB_Device\SLCparam_COMM_BattMaxP=aParam[213];
END
IF bState_Param[214]=1 THEN
    NRGLAB_Device\SLCparam_COMM_BattMinP=aParam[214];
END

IF bState_Param[228]=1 THEN
    NRGLAB_Device\SLCparam_RVM_KP=aParam[228];
END
IF bState_Param[229]=1 THEN
    NRGLAB_Device\SLCparam_RVM_KI=aParam[229];
END

```

```

END
IF bState_Param[230]=1 THEN
    NRGLAB_Device\SLCparam_RVM_MaxRate=aParam[230];
END
IF bState_Param[231]=1 THEN
    NRGLAB_Device\SLCparam_RVM_MaxDeviation=aParam[231];
END
IF bState_Param[232]=1 THEN
    NRGLAB_Device\SLCparam_RVM_MaxReconnection=aParam[232];
END
IF bState_Param[233]=1 THEN
    NRGLAB_Device\SLCparam_RVM_Ts=aParam[233];
END
IF bState_Param[234]=1 THEN
    NRGLAB_Device\SLCparam_RVM_LF_LPF=aParam[234];
END
IF bState_Param[235]=1 THEN
    NRGLAB_Device\SLCparam_RVM_HF_HPF=aParam[235];
END
IF bState_Param[236]=1 THEN
    NRGLAB_Device\SLCparam_RVM_Limit_Max=aParam[236];
END
IF bState_Param[237]=1 THEN
    NRGLAB_Device\SLCparam_RVM_Limit_Min=aParam[237];
END

IF bState_Param[244]=1 THEN
    NRGLAB_Device\SLCparam_PSM_KP=aParam[244];
END
IF bState_Param[245]=1 THEN
    NRGLAB_Device\SLCparam_PSM_KI=aParam[245];
END
IF bState_Param[246]=1 THEN
    NRGLAB_Device\SLCparam_PSM_InputRef=aParam[246];
END
IF bState_Param[247]=1 THEN
    NRGLAB_Device\SLCparam_PSM_Ts=aParam[247];
END
IF bState_Param[248]=1 THEN
    NRGLAB_Device\SLCparam_PSM_Limit_Max=aParam[248];
END
IF bState_Param[249]=1 THEN
    NRGLAB_Device\SLCparam_PSM_Limit_Min=aParam[249];
END

IF bState_Param[260]=1 THEN
    NRGLAB_Device\SLCparam_RRE_KP=aParam[260];
END
IF bState_Param[261]=1 THEN
    NRGLAB_Device\SLCparam_RRE_KI=aParam[261];
END
IF bState_Param[262]=1 THEN
    NRGLAB_Device\SLCparam_RRE_MaxRate=aParam[262];
END
IF bState_Param[263]=1 THEN
    NRGLAB_Device\SLCparam_RRE_RateLimited=aParam[263];
END
IF bState_Param[264]=1 THEN
    NRGLAB_Device\SLCparam_RRE_Ts=aParam[264];
END
IF bState_Param[265]=1 THEN
    NRGLAB_Device\SLCparam_RRE_SamplingFrecuency=aParam[265];
END
IF bState_Param[266]=1 THEN
    NRGLAB_Device\SLCparam_RRE_Limit_Max=aParam[266];
END
IF bState_Param[267]=1 THEN
    NRGLAB_Device\SLCparam_RRE_Limit_Min=aParam[267];
END

IF bState_Param[276]=1 THEN
    NRGLAB_Device\SLCparam_PCM_KP=aParam[276];
END
IF bState_Param[277]=1 THEN
    NRGLAB_Device\SLCparam_PCM_KI=aParam[277];
END

```

```

IF bState_Param[278]=1 THEN
    NRGLAB_Device\SLCparam_PCM_BalanceControlRef=aParam[278];
END
IF bState_Param[279]=1 THEN
    NRGLAB_Device\SLCparam_PCM_DeltaControlRef=aParam[279];
END
IF bState_Param[280]=1 THEN
    NRGLAB_Device\SLCparam_PCM_StopRegulationP=aParam[280];
END
IF bState_Param[281]=1 THEN
    NRGLAB_Device\SLCparam_PCM_Ts=aParam[281];
END
IF bState_Param[282]=1 THEN
    NRGLAB_Device\SLCparam_PCM_Limit_Max=aParam[282];
END
IF bState_Param[283]=1 THEN
    NRGLAB_Device\SLCparam_PCM_Limit_Min=aParam[283];
END

```

```

IF bState_Param[292]=1 THEN
    NRGLAB_Device\SLCparam_PBP_KP=aParam[292];
END
IF bState_Param[293]=1 THEN
    NRGLAB_Device\SLCparam_PBP_KI=aParam[293];
END
IF bState_Param[294]=1 THEN
    NRGLAB_Device\SLCparam_PBP_LinePowerReference=aParam[294];
END
IF bState_Param[295]=1 THEN
    NRGLAB_Device\SLCparam_PBP_FrequencyReference=aParam[295];
END
IF bState_Param[296]=1 THEN
    NRGLAB_Device\SLCparam_PBP_Ts=aParam[296];
END
IF bState_Param[297]=1 THEN
    NRGLAB_Device\SLCparam_PBP_Limit_Max=aParam[297];
END
IF bState_Param[298]=1 THEN
    NRGLAB_Device\SLCparam_PBP_Limit_Min=aParam[298];
END

```

END

//Passing PLC tags to aParam

FUNCTION loadParameters()

```

IF bState_Param[20]=0 THEN
    aParam[20]=(NRGLAB_Device\FLCparamMB_COMM_NominalPower)/1000;
END
IF bState_Param[21]=0 THEN
    aParam[21]=NRGLAB_Device\FLCparamMB_COMM_NominalVoltage;
END
IF bState_Param[22]=0 THEN
    aParam[22]=NRGLAB_Device\FLCparamMB_COMM_NominalFreq;
END
IF bState_Param[23]=0 THEN
    aParam[23]=(NRGLAB_Device\FLCparamMB_COMM_ActivePowerLimit)/1000;
END
IF bState_Param[24]=0 THEN
    aParam[24]=(NRGLAB_Device\FLCparamMB_COMM_ReactivePowerLim)/1000;
END

```

```

IF bState_Param[36]=0 THEN
    aParam[36]=NRGLAB_Device\FLCparamMB_IE_Inertia_Constant;
END
IF bState_Param[37]=0 THEN
    aParam[37]=NRGLAB_Device\FLCparamMB_IE_HighPassFiltConst;
END
IF bState_Param[38]=0 THEN
    aParam[38]=NRGLAB_Device\FLCparamMB_IE_Max_ROCOF;
END
IF bState_Param[39]=0 THEN
    aParam[39]=(NRGLAB_Device\FLCparamMB_IE_PowerLimit_Inertia)/1000;
END

```

```

IF bState_Param[52]=0 THEN
    aParam[52]=NRGLAB_Device\FLCparamMB_FR_DroopConstantP;
END
IF bState_Param[53]=0 THEN
    aParam[53]=(NRGLAB_Device\FLCparamMB_FR_Max_delta_Energy)/1000;
END
IF bState_Param[54]=0 THEN
    aParam[54]=(NRGLAB_Device\FLCparamMB_FR_PowerLimit_Freq)/1000;
END
IF bState_Param[55]=0 THEN
    aParam[55]=NRGLAB_Device\FLCparamMB_FR_FrequencyCutOff;
END
IF bState_Param[56]=0 THEN
    aParam[56]=NRGLAB_Device\FLCparamMB_FR_FreqDeadBand;
END

```

```

IF bState_Param[69]=0 THEN
    aParam[69]=NRGLAB_Device\FLCparamMB_MFA_Gain;
END
IF bState_Param[70]=0 THEN
    aParam[70]=NRGLAB_Device\FLCparamMB_MFA_BPF_CenterFreq;
END
IF bState_Param[71]=0 THEN
    aParam[71]=(NRGLAB_Device\FLCparamMB_MFA_PowerLimit_MF)/1000;
END

```

```

IF bState_Param[84]=0 THEN
    aParam[84]=NRGLAB_Device\FLCparamMB_VR_DroopConstantV;
END
IF bState_Param[85]=0 THEN
    aParam[85]=(NRGLAB_Device\FLCparamMB_VR_PowerLimit_Voltage)/1000;
END
IF bState_Param[86]=0 THEN
    aParam[86]=NRGLAB_Device\FLCparamMB_VR_VoltageCutOff;
END
IF bState_Param[87]=0 THEN
    aParam[87]=NRGLAB_Device\FLCparamMB_VR_VoltDeadBand;
END

```

```

//-----//
//FLC: EAP Communication
//-----//

```

```

IF bState_Param[116]=0 THEN
    aParam[116]=(NRGLAB_Device\FLCparamEAP_COMM_NominalPower)/1000;
END
IF bState_Param[117]=0 THEN
    aParam[117]=NRGLAB_Device\FLCparamEAP_COMM_NominalFreq;
END
IF bState_Param[118]=0 THEN
    aParam[118]=NRGLAB_Device\FLCparamEAP_COMM_NominalVolt;
END
IF bState_Param[119]=0 THEN
    aParam[119]=(NRGLAB_Device\FLCparamEAP_COMM_ActivePowLim)/1000;
END
IF bState_Param[120]=0 THEN
    aParam[120]=(NRGLAB_Device\FLCparamEAP_COMM_ReactivePowLim)/1000;
END
IF bState_Param[121]=0 THEN
    aParam[121]=NRGLAB_Device\FLCparamEAP_COMM_Delta_limits;
END
IF bState_Param[122]=0 THEN
    aParam[122]=NRGLAB_Device\FLCparamEAP_COMM_E_Limits;
END
IF bState_Param[123]=0 THEN
    aParam[123]=NRGLAB_Device\FLCparamEAP_COMM_Kp_ReactiveCtrl;
END
IF bState_Param[124]=0 THEN
    aParam[124]=NRGLAB_Device\FLCparamEAP_COMM_Ki_ReactiveCtrl;
END
IF bState_Param[125]=0 THEN
    aParam[125]=NRGLAB_Device\FLCparamEAP_COMM_VirtReactance;
END

```

```

END
IF bState_Param[126]=0 THEN
    aParam[126]=NRGLAB_Device\FLCparamEAP_COMM_VirtResistor;
END

IF bState_Param[132]=0 THEN
    aParam[132]=NRGLAB_Device\FLCparamEAP_IE_Inertia_Constant;
END
IF bState_Param[133]=0 THEN
    aParam[133]=NRGLAB_Device\FLCparamEAP_IE_Damping_Constant;
END
IF bState_Param[134]=0 THEN
    aParam[134]=NRGLAB_Device\FLCparamEAP_IE_HighPassFiltConst;
END
IF bState_Param[135]=0 THEN
    aParam[135]=NRGLAB_Device\FLCparamEAP_IE_Max_ROCOF;
END

IF bState_Param[148]=0 THEN
    aParam[148]=NRGLAB_Device\FLCparamEAP_FR_DroopConstantP;
END
IF bState_Param[149]=0 THEN
    aParam[149]=(NRGLAB_Device\FLCparamEAP_FR_Max_Delta_Energy)/1000;
END
IF bState_Param[150]=0 THEN
    aParam[150]=(NRGLAB_Device\FLCparamEAP_FR_PowerLimit_Freq)/1000;
END
IF bState_Param[151]=0 THEN
    aParam[151]=NRGLAB_Device\FLCparamEAP_FR_LPF_CutOff_Freq;
END
IF bState_Param[152]=0 THEN
    aParam[152]=(NRGLAB_Device\FLCparamEAP_FR_DeadBand_Limits_P)/1000;
END

IF bState_Param[164]=0 THEN
    aParam[164]=NRGLAB_Device\FLCparamEAP_MFA_Gain;
END
IF bState_Param[165]=0 THEN
    aParam[165]=(NRGLAB_Device\FLCparamEAP_MFA_PowerLimit)/1000;
END
IF bState_Param[166]=0 THEN
    aParam[166]=NRGLAB_Device\FLCparamEAP_MFA_Max_ROCOF_Modal;
END
IF bState_Param[167]=0 THEN
    aParam[167]=NRGLAB_Device\FLCparamEAP_MFA_BPF_CenterFreq;
END

IF bState_Param[180]=0 THEN
    aParam[180]=NRGLAB_Device\FLCparamEAP_VR_DroopConstantQ;
END
IF bState_Param[181]=0 THEN
    aParam[181]=NRGLAB_Device\FLCparamEAP_VR_LPF_CutOff_Volt;
END
IF bState_Param[182]=0 THEN
    aParam[182]=NRGLAB_Device\FLCparamEAP_VR_DeadBand_Limit_Q;
END
IF bState_Param[183]=0 THEN
    aParam[183]=(NRGLAB_Device\FLCparamEAP_VR_PowerLimit_Volt)/1000;
END

```

//-----//

//SLC Communication

//-----//

```

IF bState_Param[212]=0 THEN
    aParam[212]=NRGLAB_Device\SLCparam_COMM_BattSoC;
END
IF bState_Param[213]=0 THEN
    aParam[213]=NRGLAB_Device\SLCparam_COMM_BattMaxP;

```



```

END
IF bState_Param[214]=0 THEN
    aParam[214]=NRGLAB_Device\SLCparam_COMM_BattMinP;
END

IF bState_Param[228]=0 THEN
    aParam[228]=NRGLAB_Device\SLCparam_RVM_KP;
END
IF bState_Param[229]=0 THEN
    aParam[229]=NRGLAB_Device\SLCparam_RVM_KI;
END
IF bState_Param[230]=0 THEN
    aParam[230]=NRGLAB_Device\SLCparam_RVM_MaxRate;
END
IF bState_Param[231]=0 THEN
    aParam[231]=NRGLAB_Device\SLCparam_RVM_MaxDeviation;
END
IF bState_Param[232]=0 THEN
    aParam[232]=NRGLAB_Device\SLCparam_RVM_MaxReconnection;
END
IF bState_Param[233]=0 THEN
    aParam[233]=NRGLAB_Device\SLCparam_RVM_Ts;
END
IF bState_Param[234]=0 THEN
    aParam[234]=NRGLAB_Device\SLCparam_RVM_LF_LPF;
END
IF bState_Param[235]=0 THEN
    aParam[235]=NRGLAB_Device\SLCparam_RVM_HF_HPF;
END
IF bState_Param[236]=0 THEN
    aParam[236]=NRGLAB_Device\SLCparam_RVM_Limit_Max;
END
IF bState_Param[237]=0 THEN
    aParam[237]=NRGLAB_Device\SLCparam_RVM_Limit_Min;
END

IF bState_Param[244]=0 THEN
    aParam[244]=NRGLAB_Device\SLCparam_PSM_KP;
END
IF bState_Param[245]=0 THEN
    aParam[245]=NRGLAB_Device\SLCparam_PSM_KI;
END
IF bState_Param[246]=0 THEN
    aParam[246]=NRGLAB_Device\SLCparam_PSM_InputRef;
END
IF bState_Param[247]=0 THEN
    aParam[247]=NRGLAB_Device\SLCparam_PSM_Ts;
END
IF bState_Param[248]=0 THEN
    aParam[248]=NRGLAB_Device\SLCparam_PSM_Limit_Max;
END
IF bState_Param[249]=0 THEN
    aParam[249]=NRGLAB_Device\SLCparam_PSM_Limit_Min;
END

IF bState_Param[260]=0 THEN
    aParam[260]=NRGLAB_Device\SLCparam_RRE_KP;
END
IF bState_Param[261]=0 THEN
    aParam[261]=NRGLAB_Device\SLCparam_RRE_KI;
END
IF bState_Param[262]=0 THEN
    aParam[262]=NRGLAB_Device\SLCparam_RRE_MaxRate;
END
IF bState_Param[263]=0 THEN
    aParam[263]=NRGLAB_Device\SLCparam_RRE_RateLimited;
END
IF bState_Param[264]=0 THEN
    aParam[264]=NRGLAB_Device\SLCparam_RRE_Ts;
END
IF bState_Param[265]=0 THEN
    aParam[265]=NRGLAB_Device\SLCparam_RRE_SamplingFrequency;
END
IF bState_Param[266]=0 THEN
    aParam[266]=NRGLAB_Device\SLCparam_RRE_Limit_Max;
END
IF bState_Param[267]=0 THEN
    aParam[267]=NRGLAB_Device\SLCparam_RRE_Limit_Min;

```

END

```
IF bState_Param[276]=0 THEN
    aParam[276]=NRGLAB_Device\SLCparam_PCM_KP;
END
IF bState_Param[277]=0 THEN
    aParam[277]=NRGLAB_Device\SLCparam_PCM_KI;
END
IF bState_Param[278]=0 THEN
    aParam[278]=NRGLAB_Device\SLCparam_PCM_BalanceControlRef;
END
IF bState_Param[279]=0 THEN
    aParam[279]=NRGLAB_Device\SLCparam_PCM_DeltaControlRef;
END
IF bState_Param[280]=0 THEN
    aParam[280]=NRGLAB_Device\SLCparam_PCM_StopRegulationP;
END
IF bState_Param[281]=0 THEN
    aParam[281]=NRGLAB_Device\SLCparam_PCM_Ts;
END
IF bState_Param[282]=0 THEN
    aParam[282]=NRGLAB_Device\SLCparam_PCM_Limit_Max;
END
IF bState_Param[283]=0 THEN
    aParam[283]=NRGLAB_Device\SLCparam_PCM_Limit_Min;
END
```

```
IF bState_Param[292]=0 THEN
    aParam[292]=NRGLAB_Device\SLCparam_PBP_KP;
END
IF bState_Param[293]=0 THEN
    aParam[293]=NRGLAB_Device\SLCparam_PBP_KI;
END
IF bState_Param[294]=0 THEN
    aParam[294]=NRGLAB_Device\SLCparam_PBP_LinePowerReference;
END
IF bState_Param[295]=0 THEN
    aParam[295]=NRGLAB_Device\SLCparam_PBP_FrequencyReference;
END
IF bState_Param[296]=0 THEN
    aParam[296]=NRGLAB_Device\SLCparam_PBP_Ts;
END
IF bState_Param[297]=0 THEN
    aParam[297]=NRGLAB_Device\SLCparam_PBP_Limit_Max;
END
IF bState_Param[298]=0 THEN
    aParam[298]=NRGLAB_Device\SLCparam_PBP_Limit_Min;
END
```

END

```

//#####//
//Functions used in Parametrization Screens for Commands
//In general Commands are converted to Digital values (see "Digital Integer Conversion.ci")
//#####//

GLOBAL INT bState_Comand[20];    ! This array indicates if the command has been changes

//#####//

//Change bit value (toggle)
FUNCTION Change_ComandBit(INT i, INT iBit)
    BoolToInt(i,iBit);
END

//All State Commands and State bit Commands to '0'
FUNCTION Reset_StateComand()
    INT i;
    FOR i=0 TO 20 DO
        Reset_ServiceComand(i);
    END
END

//Reset Service Command by turning bStateBit_Comand to 0
//By this way, SCADA is reading and showing PLC command again
FUNCTION Reset_ServiceComand(INT i)
    INT j=0;
    bState_Comand[i]=0;
    WHILE j<16 DO
        bStateBit_Comand[i][j]=0;
        j=j+1;
    END
END

//Update all values from Commands
FUNCTION Update_Comand()
    ComandToTagUpdate();
    SleepMS(50);
    Reset_StateComand();
END

//For PCM service, only one bit (a part from enable/unable) can be active at the same Time.
//When one bit IS activated, other have TO be deactivated
FUNCTION ExcpesionCommand(INT i, INT j)
    IF i=17 THEN
        SELECT CASE j
            CASE 1
                IF aDigitalValue[i][2]=1 THEN
                    aDigitalValue[i][2]=0;
                    bStateBit_Comand[i][2]=1;
                END IF
            END SELECT
        END IF
    END IF
END

```

```

END

IF aDigitalValue[i][3]=1 THEN
    aDigitalValue[i][3]=0;
    bStateBit_Comand[i][3]=1;

END

IF aDigitalValue[i][4]=1 THEN
    aDigitalValue[i][4]=0;
    bStateBit_Comand[i][4]=1;

END

CASE 2

IF aDigitalValue[i][3]=1 THEN
    aDigitalValue[i][3]=0;
    bStateBit_Comand[i][3]=1;

END

IF aDigitalValue[i][4]=1 THEN
    aDigitalValue[i][4]=0;
    bStateBit_Comand[i][4]=1;

END

IF aDigitalValue[i][1]=1 THEN
    aDigitalValue[i][1]=0;
    bStateBit_Comand[i][1]=1;

END

CASE 3

IF aDigitalValue[i][4]=1 THEN
    aDigitalValue[i][4]=0;
    bStateBit_Comand[i][4]=1;

END

IF aDigitalValue[i][1]=1 THEN
    aDigitalValue[i][1]=0;
    bStateBit_Comand[i][1]=1;

END

IF aDigitalValue[i][2]=1 THEN
    aDigitalValue[i][2]=0;
    bStateBit_Comand[i][2]=1;

END

CASE 4

IF aDigitalValue[i][1]=1 THEN
    aDigitalValue[i][1]=0;
    bStateBit_Comand[i][1]=1;

END

IF aDigitalValue[i][2]=1 THEN
    aDigitalValue[i][2]=0;
    bStateBit_Comand[i][2]=1;

END

IF aDigitalValue[i][3]=1 THEN
    aDigitalValue[i][3]=0;
    bStateBit_Comand[i][3]=1;

```

```

        END

    END SELECT

END

END

//The aim of this function is to set on Delta Control Curtailment bit on in PCM when SCADA start to run

FUNCTION ForceCommandPCM()

    aDigitalValue[17][3]=1;
    ExcepcionCommand(17,3);
    aComand[17]=Int_Conversion(17);
    NRGLAB_Device\SLCcmd_PCM=aComand[17];

END

//For those Commands that have been changed, the new value is passed to PLC Command

FUNCTION ComandToTagUpdate()

    IF bState_Comand[1]=1 THEN
        NRGLAB_Device\FLCcmdMB_COMM=aComand[1];
    END
    IF bState_Comand[2]=1 THEN
        NRGLAB_Device\FLCcmdMB_IE=aComand[2];
    END
    IF bState_Comand[3]=1 THEN
        NRGLAB_Device\FLCcmdMB_FR=aComand[3];
    END
    IF bState_Comand[4]=1 THEN
        NRGLAB_Device\FLCcmdMB_MFA=aComand[4];
    END
    IF bState_Comand[5]=1 THEN
        NRGLAB_Device\FLCcmdMB_VR=aComand[5];
    END

    IF bState_Comand[7]=1 THEN
        NRGLAB_Device\FLCcmdEAP_COMM=aComand[7];
    END
    IF bState_Comand[8]=1 THEN
        NRGLAB_Device\FLCcmdEAP_IE=aComand[8];
    END
    IF bState_Comand[9]=1 THEN
        NRGLAB_Device\FLCcmdEAP_FR=aComand[9];
    END
    IF bState_Comand[10]=1 THEN
        NRGLAB_Device\FLCcmdEAP_MFA=aComand[10];
    END
    IF bState_Comand[11]=1 THEN
        NRGLAB_Device\FLCcmdEAP_VR=aComand[11];
    END

    IF bState_Comand[13]=1 THEN
        NRGLAB_Device\SLCcmd_COMM=aComand[13];
    END
    IF bState_Comand[14]=1 THEN
        NRGLAB_Device\SLCcmd_RVM=aComand[14];
    END
    IF bState_Comand[15]=1 THEN
        NRGLAB_Device\SLCcmd_PSM=aComand[15];
    END
    IF bState_Comand[16]=1 THEN
        NRGLAB_Device\SLCcmd_RRE=aComand[16];
    END
    IF bState_Comand[17]=1 THEN
        NRGLAB_Device\SLCcmd_PCM=aComand[17];
    END
    IF bState_Comand[18]=1 THEN
        NRGLAB_Device\SLCcmd_PBP=aComand[18];
    END

END

//If bState_Comand[i]=0, not changed, SCADA command gets its value from PLC

```

```

FUNCTION loadComand()

    IF bState_Comand[1]=0 THEN
        aComand[1]=NRGLAB_Device\FLCcmdMB_COMM;
        IntToBool(1);
    END
    IF bState_Comand[2]=0 THEN
        aComand[2]=NRGLAB_Device\FLCcmdMB_IE;
        IntToBool(2);
    END
    IF bState_Comand[3]=0 THEN
        aComand[3]=NRGLAB_Device\FLCcmdMB_FR;
        IntToBool(3);
    END
    IF bState_Comand[4]=0 THEN
        aComand[4]=NRGLAB_Device\FLCcmdMB_MFA;
        IntToBool(4);
    END
    IF bState_Comand[5]=0 THEN
        aComand[5]=NRGLAB_Device\FLCcmdMB_VR;
        IntToBool(5);
    END

    IF bState_Comand[7]=0 THEN
        aComand[7]=NRGLAB_Device\FLCcmdEAP_COMM;
    END
    IF bState_Comand[8]=0 THEN
        aComand[8]=NRGLAB_Device\FLCcmdEAP_IE;
        IntToBool(8);
    END
    IF bState_Comand[9]=0 THEN
        aComand[9]=NRGLAB_Device\FLCcmdEAP_FR;
        IntToBool(9);
    END
    IF bState_Comand[10]=0 THEN
        aComand[10]=NRGLAB_Device\FLCcmdEAP_MFA;
        IntToBool(10);
    END
    IF bState_Comand[11]=0 THEN
        aComand[11]=NRGLAB_Device\FLCcmdEAP_VR;
        IntToBool(11);
    END

    IF bState_Comand[13]=0 THEN
        aComand[13]=NRGLAB_Device\SLCcmd_COMM;
        IntToBool(13);
    END
    IF bState_Comand[14]=0 THEN
        aComand[14]=NRGLAB_Device\SLCcmd_RVM;
        IntToBool(14);
    END
    IF bState_Comand[15]=0 THEN
        aComand[15]=NRGLAB_Device\SLCcmd_PSM;
        IntToBool(15);
    END
    IF bState_Comand[16]=0 THEN
        aComand[16]=NRGLAB_Device\SLCcmd_RRE;
        IntToBool(16);
    END
    IF bState_Comand[17]=0 THEN
        aComand[17]=NRGLAB_Device\SLCcmd_PCM;
        IntToBool(17);
    END
    IF bState_Comand[18]=0 THEN
        aComand[18]=NRGLAB_Device\SLCcmd_PBP;
        IntToBool(18);
    END

END

```

```

//#####//
// This file contains functions used to convert integer to digital values and viceversa
//#####//

GLOBAL INT aDigitalAnterior[20]; ! Array used to store the previous value of the int number. Used to avoid any
porblem that has been seen during testing

//#####//

//By this function the Integer value stored in the array is converted to Digital
FUNCTION IntToBool(INT i)

    INT j=0;
    INT b=1;
    INT n=aComand[i];    !Get Command value

    IF aDigitalAnterior[i]<>n THEN    ! Done to avoid problems of overlap seen during testing
        n=0;
        b=0;
    END

    aDigitalAnterior[i]=aComand[i];

    WHILE j<16 AND b=1 DO    !Procedure to convert an integer value to its digital form

        aDigitalValue[i][j]=n MOD 2;
        n=n/2;
        j=j+1;

    END

    b=1;

END

//By this function the Digital value is changed in one bit and then this change is passed to the Integer value
FUNCTION BoolToInt(INT i, INT iBit)

    INT n;

    bState_Comand[i]=1;    !State Command turns to 1, indicating that Command it has been changed
    bStateBit_Comand[i][iBit]=1; !State Bit Command turns to 1, indicating that this bit has been changed

    Toggle(aDigitalValue[i][iBit]);    !Change value of command bit (0->1 or 1->0)

    ExcepcionCommand(i,iBit);    !In case it's been changed PCM command

    n=Int_Conversion(i); !Update aComand[i] (integer) with this modification

    aComand[i]=n;

END

//Convert a digital value to integer
INT FUNCTION Int_Conversion(INT i)

    INT n=0;
    INT j;

    WHILE j<16 DO
        n = n + aDigitalValue[i][j]*(Pow(2,j));
        j=j+1;
    END

    RETURN n;

END

```

```

//#####//
/*Functions used in Parametrization Screens for Maximum Values      */
//#####//

//All aMax positions get their values from aMax_initial values stored in the program

FUNCTION Transfer_Maximum()

    INT i;

    FOR i=0 TO 323 DO

        IF bState_Max[i]=0 THEN
            aMax[i]=aMax_initial[i];
        END

    END

END

//Change Maximum value from a variable
//Value will be changed, activating bState_Max[i] bit

FUNCTION Change_Maximum(INT i, STRING title)

    REAL value;
    STRING new;
    STRING ongoing=RealToStr(aMax[i],8,3,".");    ! Convert Real value to String to have appropriate format for Input
function

    new=Input(title, "Value", ongoing);           ! Input function where a pop-up box opens and the current value is
displayed and can be changed

    IF IsError()=299 THEN    ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

    value=StrToReal(new);    ! Convert String value result to Real again

    IF value <> aMax[i] THEN ! Only if value have changed
        Change_StateMaximum(i);
    END

    aMax[i]=value;

END

//bState_Max[i] turns to 1 due to a change on the maximum value

FUNCTION Change_StateMaximum(INT i)

    bState_Max[i]=1;

END

//Turn all bState_Max[i] to 0

FUNCTION Reset_StateMaximum()

    INT i;

    FOR i=0 TO 323 DO
        bState_Max[i]=0;
    END

END

//Turn all bState_Max[i] from a service to 0

FUNCTION Reset_ServiceMaximum(INT i, INT j)

    FOR i=i TO j DO
        bState_Max[i]=0;
    END

END

```


END

//Store all new values of aMax to aMax_initial and turn all bState_Max[i] to 0

FUNCTION Update_Maximum()

INT i;

FOR i=0 TO 323 DO

aMax_initial[i]=aMax[i];

END

Reset_StateMaximum();

END

//Undo all changes done on maximumvalues, by turning all bState_Max[i] to 0 and
//getting all values from aMax_initial again.

FUNCTION Undo_Maximum()

Reset_StateMaximum();

Transfer_Maximum();

END

//-----//
//By this function is possible to read all Maximum values from a local SQL Data Base

//FUNCTION SQL_Maximum()

//

// INT hSQL;

// INT i=0;

// INT ok;

//

// hSQL=SQLConnect("SCADA Data Provider=OleDb; Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;User ID=ROGER BAUTISTA FLORENSA;Initial Catalog=SEER;Data Source=DESKTOP-P5OJOI7\ION");

// ok=SQLExec(hSQL, "Select MaxValue from MaximumParameters");

//

// IF ok=0 THEN

// WHILE SQLNext(hSQL)=0 AND i<324 DO

// aSQLMaximum[i]=SQLGetField(hSQL,"MaxValue");

// aMax_initial[i]=StrToReal(aSQLMaximum[i]);

// i=i+1;

// END

//

// SQLDisconnect(hSQL);

// END

//

//END

//-----//

//load initial values from Maximum when SCADA is started

FUNCTION loadMaximum()

aMax_initial[20] =1000; !It's 1E6 but there is no way to fit it (units changed)

aMax_initial[21] = 400;

aMax_initial[22] = 60;

aMax_initial[23] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aMax_initial[24] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aMax_initial[36] = 20;

aMax_initial[37] = 10;

aMax_initial[38] = 10;

aMax_initial[39] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aMax_initial[52] = 10;

aMax_initial[53] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aMax_initial[54] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aMax_initial[55] = 10;

aMax_initial[56] = 100;

aMax_initial[69] = 2000;

aMax_initial[70] = 50;

aMax_initial[71] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aMax_initial[84] = 100;

aMax_initial[84] = 1000; !It's 1E6 but there is no way to fit it (units changed)

```

aMax_initial[86] = 50;
aMax_initial[87] = 400;

aMax_initial[116] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aMax_initial[117] = 60;
aMax_initial[118] = 400;
aMax_initial[119] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aMax_initial[120] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aMax_initial[121] = 10;
aMax_initial[122] = 500;
aMax_initial[123] = 100;
aMax_initial[124] = 10;
aMax_initial[125] = 1;
aMax_initial[126] = 1;

aMax_initial[132] = 20;
aMax_initial[133] = 10;
aMax_initial[134] = 10;
aMax_initial[135] = 10;

aMax_initial[148] = 100;
aMax_initial[149] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aMax_initial[150] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aMax_initial[151] = 100;
aMax_initial[152] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aMax_initial[164] = 2000;
aMax_initial[165] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aMax_initial[166] = 100;
aMax_initial[167] = 10;

aMax_initial[180] = 100;
aMax_initial[181] = 50;
aMax_initial[182] = 400;
aMax_initial[183] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aMax_initial[212] = 100;
aMax_initial[213] = 1;
aMax_initial[214] = 1;

aMax_initial[228] = 10;
aMax_initial[229] = 10;
aMax_initial[230] = 1;
aMax_initial[231] = 1;
aMax_initial[232] = 1;
aMax_initial[233] = 1;
aMax_initial[234] = 10;
aMax_initial[235] = 20;
aMax_initial[236] = 1;
aMax_initial[237] = 1;

aMax_initial[244] = 10;
aMax_initial[245] = 10;
aMax_initial[246] = 1;
aMax_initial[247] = 1;
aMax_initial[248] = 1;
aMax_initial[249] = 1;

aMax_initial[260] = 10;
aMax_initial[261] = 10;
aMax_initial[262] = 1;
aMax_initial[263] = 1;
aMax_initial[264] = 1;
aMax_initial[265] = 10000;
aMax_initial[266] = 1;
aMax_initial[267] = 1;

aMax_initial[276] = 10;
aMax_initial[277] = 10;
aMax_initial[278] = 1;
aMax_initial[279] = 1;
aMax_initial[280] = 1;
aMax_initial[281] = 1;
aMax_initial[282] = 1;
aMax_initial[283] = 1;

aMax_initial[292] = 10;
aMax_initial[293] = 10;

```

```
aMax_initial[294] = 1;  
aMax_initial[295] = 60;  
aMax_initial[296] = 1;  
aMax_initial[297] = 1;  
aMax_initial[298] = 1;
```

```
END
```

```

//#####//
/*Functions used in Parametrization Screens for Minimum Values          */
//#####//

//All aMin positions get their values from aMin_initial values stored in the program

FUNCTION Transfer_Minimum()

    INT i;

    FOR i=0 TO 323 DO

        IF bState_Min[i]=0 THEN
            aMin[i]=aMin_initial[i];
        END

    END

END

//Change Minimum value from a variable
//Value will be changed, activating bState_Min[i] bit

FUNCTION Change_Minimum(INT i, STRING title)

    REAL value;
    STRING new;
    STRING ongoing=RealToStr(aMin[i],8,3,".");    ! Convert Real value to String to have appropriate format for Input
function

    new=Input(title, "Value", ongoing);           ! Input function where a pop-up box opens and the current value is
displayed and can be changed

    IF IsError()=299 THEN    ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

    value=StrToReal(new);    ! Convert String value result to Real again

    IF value <> aMin[i] THEN ! Only if value have changed
        Change_StateMinimum(i);
    END

    aMin[i]=value;
END

//bState_MIN[i] turns to 1 due to a change on the minimum value

FUNCTION Change_StateMinimum(INT i)

    bState_Min[i]=1;

END

//Turn all bState_Min[i] to

FUNCTION Reset_StateMinimum()

    INT i;

    FOR i=0 TO 323 DO
        bState_Min[i]=0;
    END

END

//Turn all bState_Min[i] from a service to 0

FUNCTION Reset_ServiceMinimum(INT i, INT j)

    FOR i=i TO j DO
        bState_Min[i]=0;
    END

END

```

END

//Store all new values of aMin to aMin_initial and turn all bState_Min[i] to 0

FUNCTION Update_Minimum()

INT i;

FOR i=0 TO 323 DO

aMin_initial[i]=aMin[i];

END

Reset_StateMinimum();

END

//Undo all changes done on minimum values, by turning all bState_Min[i] to 0 and
//getting all values from aMin_initial again.

FUNCTION Undo_Minimum()

Reset_StateMinimum();

Transfer_Minimum();

END

//-----//
//By this function is possible to read all Minimum values from a local SQL Data Base

//FUNCTION SQL_Minimum()

//

// INT hSQL;

// INT i=0;

// INT ok;

//

// hSQL=SQLConnect("SCADA Data Provider=OleDb; Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;User ID=ROGER BAUTISTA FLORENSA;Initial Catalog=SEER;Data Source=DESKTOP-P5OJOI7\ION");

// ok=SQLExec(hSQL, "Select MinValue from MinimumParameters");

//

// IF ok=0 THEN

// WHILE SQLNext(hSQL)=0 AND i<324 DO

// aSQLMinimum[i]=SQLGetField(hSQL,"MinValue");

// aMin_initial[i]=StrToReal(aSQLMinimum[i]);

// i=i+1;

// END

//

// SQLDisconnect(hSQL);

// END

//

//END

//-----//

//load initial values from Minimum when SCADA is started

FUNCTION loadMinimum()

aMin_initial[20] = -1000; !It's 1E6 but there is no way to fit it (units changed)

aMin_initial[21] = 0;

aMin_initial[22] = 0;

aMin_initial[23] = 0;

aMin_initial[24] = 0;

aMin_initial[36] = 0;

aMin_initial[37] = -10;

aMin_initial[38] = -10;

aMin_initial[39] = 0;

aMin_initial[52] = 0;

aMin_initial[53] = -1000; !It's 1E6 but there is no way to fit it (units changed)

aMin_initial[54] = 0;

aMin_initial[55] = 10;

aMin_initial[56] = 0;

aMin_initial[69] = 0;

aMin_initial[70] = 0;

aMin_initial[71] = 0;

aMin_initial[84] = 0;

aMin_initial[84] = 0;

```
aMin_initial[86] = 0;
aMin_initial[87] = 0;

aMin_initial[116] = 0;
aMin_initial[117] = 0;
aMin_initial[118] = 0;
aMin_initial[119] = 0;
aMin_initial[120] = 0;
aMin_initial[121] = -10;
aMin_initial[122] = 0;
aMin_initial[123] = -10;
aMin_initial[124] = -10;
aMin_initial[125] = 0;
aMin_initial[126] = 0;

aMin_initial[132] = 0;
aMin_initial[133] = 0;
aMin_initial[134] = 0;
aMin_initial[135] = 0;

aMin_initial[148] = 0;
aMin_initial[149] = 0;
aMin_initial[150] = 0;
aMin_initial[151] = 0;
aMin_initial[152] = 0;

aMin_initial[164] = 0;
aMin_initial[165] = 0;
aMin_initial[166] = 0;
aMin_initial[167] = 0;

aMin_initial[180] = 0;
aMin_initial[181] = 0;
aMin_initial[182] = 0;
aMin_initial[183] = 0;

aMin_initial[212] = 0;
aMin_initial[213] = 0;
aMin_initial[214] = 0;

aMin_initial[228] = -10;
aMin_initial[229] = -10;
aMin_initial[230] = 0;
aMin_initial[231] = 0;
aMin_initial[232] = 0;
aMin_initial[233] = 0;
aMin_initial[234] = 0;
aMin_initial[235] = 0;
aMin_initial[236] = 0;
aMin_initial[237] = 0;

aMin_initial[244] = -10;
aMin_initial[245] = -10;
aMin_initial[246] = 0;
aMin_initial[247] = 0;
aMin_initial[248] = 0;
aMin_initial[249] = 0;

aMin_initial[260] = -10;
aMin_initial[261] = -10;
aMin_initial[262] = 0;
aMin_initial[263] = 0;
aMin_initial[264] = 0;
aMin_initial[265] = 0;
aMin_initial[266] = 0;
aMin_initial[267] = 0;

aMin_initial[276] = -10;
aMin_initial[277] = -10;
aMin_initial[278] = -1;
aMin_initial[279] = 0;
aMin_initial[280] = 0;
aMin_initial[281] = 0;
aMin_initial[282] = 0;
aMin_initial[283] = 0;

aMin_initial[292] = -10;
aMin_initial[293] = -10;
```

```
aMin_initial[294] = -1;  
aMin_initial[295] = 0;  
aMin_initial[296] = 0;  
aMin_initial[297] = 0;  
aMin_initial[298] = 0;  
END
```

```

//#####//
/*Functions used in Parametrization Screens for Default Values          */
//#####//

//All aDef positions get their values from aDef_initial values stored in the program

FUNCTION Transfer_Default()

    INT i;

    FOR i=0 TO 323 DO
        IF bState_Def[i]=0 THEN
            aDef[i]=aDef_initial[i];
        END
    END

END

//Change Default value from a variable
//Value will be changed, activating bState_Def[i] bit, if it is => Minimum or =< Maximum

FUNCTION Change_Default(INT i, STRING title)

    REAL value;
    STRING new;
    STRING ongoing=RealToStr(aDef[i],8,3,".");    ! Convert Real value to String to have appropriate format for Input
function

    new=Input(title, "Value", ongoing);            ! Input function where a pop-up box opens and the current value is
displayed and can be changed

    IF IsError()=299 THEN    ! If there is an error in the Input function previous value is given as response, thus
this value is passed to new value.
        new=ongoing;
    END

    value=StrToReal(new);    ! Convert String value result to Real again

    IF value <> aDef[i] THEN ! Only if value have changed

        IF value>aMax_initial[i] OR value<aMin_initial[i] THEN
            Message("Error","Entered value is out of bounds",48);
        ELSE
            Change_StateDefault(i);
            aDef[i]=value;
        END

    END

END

//bState_Def[i] turns to 1 due to a change on the default value

FUNCTION Change_StateDefault(INT i)

    bState_Def[i]=1;

END

//Turn all bState_Def[i] to 0

FUNCTION Reset_StateDefault()

    INT i;

    FOR i=0 TO 323 DO
        bState_Def[i]=0;
    END

END

//Turn all bState_Def[i] from a service to 0

FUNCTION Reset_ServiceDefault(INT i, INT j)

    FOR i=i TO j DO

```



```

        bState_Def[i]=0;
    END

END

//Store all new values of aDef to aDef_initial and turn all bState_Def[i] to 0

FUNCTION Update_Default()

    INT i;

    FOR i=0 TO 323 DO
        aDef_initial[i]=aDef[i];
    END

    Reset_StateDefault();

END

//Undo all changes done on default values, by turning all bState_Def[i] to 0 and
//getting all values from aDef_initial again.

FUNCTION Undo_Default()

    Reset_StateDefault();
    Transfer_Default();

END

//-----//
//By this function is possible to read all Default values from a local SQL Data Base
//
//FUNCTION SQL_Default()
//
//    INT hSQL;
//    INT i=0;
//    INT ok;
//
//    hSQL=SQLConnect("SCADA Data Provider=OleDb; Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;User ID=ROGER BAUTISTA FLORENSA;Initial Catalog=SEER;Data Source=DESKTOP-P5OJOI7\ION");
//    ok=SQLExec(hSQL, "Select DefValue from DefaultParameters");
//
//    IF ok=0 THEN
//        WHILE SQLNext(hSQL)=0 AND i<324 DO
//            aSQLDefault[i]=SQLGetField(hSQL,"DefValue");
//            aDef_initial[i]=StrToReal(aSQLDefault[i]);
//            i=i+1;
//        END
//
//        SQLDisconnect(hSQL);
//    END
//
//END
//-----//

//load initial values from Deafult when SCADA is started

FUNCTION loadDefault()

    aDef_initial[20] = 1000; !It's 1E6 but there is no way to fit it (units changed)
    aDef_initial[21] = 240;
    aDef_initial[22] = 50;
    aDef_initial[23] = 20;
    aDef_initial[24] = 1000; !It's 1E6 but there is no way to fit it (units changed)

    aDef_initial[36] = 5;
    aDef_initial[37] = 0.1;
    aDef_initial[38] = 5;
    aDef_initial[39] = 1000; !It's 1E6 but there is no way to fit it (units changed)

    aDef_initial[52] = 4;
    aDef_initial[53] = 1000; !It's 1E6 but there is no way to fit it (units changed)
    aDef_initial[54] = 1000; !It's 1E6 but there is no way to fit it (units changed)
    aDef_initial[55] = 0.2;
    aDef_initial[56] = 30;

    aDef_initial[69] = 1000;
    aDef_initial[70] = 31.4;
    aDef_initial[71] = 1000; !It's 1E6 but there is no way to fit it (units changed)

```

```

aDef_initial[84] = 20;
aDef_initial[84] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aDef_initial[86] = 0.1;
aDef_initial[87] = 20;

aDef_initial[116] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aDef_initial[117] = 50;
aDef_initial[118] = 240;
aDef_initial[119] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aDef_initial[120] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aDef_initial[121] = 1;
aDef_initial[122] = 200;
aDef_initial[123] = 0.0001;
aDef_initial[124] = 0.0001;
aDef_initial[125] = 0.3;
aDef_initial[126] = 0.1;

aDef_initial[132] = 5;
aDef_initial[133] = 0.7;
aDef_initial[134] = 0.1;
aDef_initial[135] = 6;

aDef_initial[148] = 4;
aDef_initial[149] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aDef_initial[150] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aDef_initial[151] = 2;
aDef_initial[152] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aDef_initial[164] = 1000;
aDef_initial[165] = 1000; !It's 1E6 but there is no way to fit it (units changed)
aDef_initial[166] = 5;
aDef_initial[167] = 31.4;

aDef_initial[180] = 5;
aDef_initial[181] = 5;
aDef_initial[182] = 20;
aDef_initial[183] = 1000; !It's 1E6 but there is no way to fit it (units changed)

aDef_initial[212] = 0;
aDef_initial[213] = 1;
aDef_initial[214] = 1;

aDef_initial[228] = -1;
aDef_initial[229] = 0;
aDef_initial[230] = 0.1;
aDef_initial[231] = 0.6;
aDef_initial[232] = 0.3;
aDef_initial[233] = 1;
aDef_initial[234] = 0.1;
aDef_initial[235] = 1;
aDef_initial[236] = 1;
aDef_initial[237] = -1;

aDef_initial[244] = 0.03;
aDef_initial[245] = 0.015;
aDef_initial[246] = 0;
aDef_initial[247] = 0.1;
aDef_initial[248] = 1;
aDef_initial[249] = -1;

aDef_initial[260] = -1;
aDef_initial[261] = 0;
aDef_initial[262] = 0.25;
aDef_initial[263] = 0;
aDef_initial[264] = 0.1;
aDef_initial[265] = 10;
aDef_initial[266] = 1;
aDef_initial[267] = -1;

aDef_initial[276] = 0;
aDef_initial[277] = 0;
aDef_initial[278] = 0;
aDef_initial[279] = 0;
aDef_initial[280] = 0;
aDef_initial[281] = 0;
aDef_initial[282] = 0;
aDef_initial[283] = 0;

```

```
aDef_initial[292] = 0;  
aDef_initial[293] = 0;  
aDef_initial[294] = 0;  
aDef_initial[295] = 0;  
aDef_initial[296] = 0;  
aDef_initial[297] = 0;  
aDef_initial[298] = 0;  
END
```

```

//#####//

//In this file there are all the functions used to pass default values to the parameters.
//Each function corresponds to one service, where are called each parameter of it.

//#####//

FUNCTION DefaultToParam_CB_MB()

    GetDeafult_Parameter(20);
    GetDeafult_Parameter(21);
    GetDeafult_Parameter(22);
    GetDeafult_Parameter(23);
    GetDeafult_Parameter(24);

END

FUNCTION DefaultToParam_IE_MB()

    GetDeafult_Parameter(36);
    GetDeafult_Parameter(37);
    GetDeafult_Parameter(38);
    GetDeafult_Parameter(39);

END

FUNCTION DefaultToParam_FR_MB()

    GetDeafult_Parameter(52);
    GetDeafult_Parameter(53);
    GetDeafult_Parameter(54);
    GetDeafult_Parameter(55);
    GetDeafult_Parameter(56);

END

FUNCTION DefaultToParam_MF_MB()

    GetDeafult_Parameter(69);
    GetDeafult_Parameter(70);
    GetDeafult_Parameter(71);

END

FUNCTION DefaultToParam_VR_MB()

    GetDeafult_Parameter(84);
    GetDeafult_Parameter(85);
    GetDeafult_Parameter(86);
    GetDeafult_Parameter(87);

END

FUNCTION DefaultToParam_CB_EAP()

    GetDeafult_Parameter(116);
    GetDeafult_Parameter(117);
    GetDeafult_Parameter(118);
    GetDeafult_Parameter(119);
    GetDeafult_Parameter(120);
    GetDeafult_Parameter(121);
    GetDeafult_Parameter(122);
    GetDeafult_Parameter(123);
    GetDeafult_Parameter(124);
    GetDeafult_Parameter(125);
    GetDeafult_Parameter(126);

END

FUNCTION DefaultToParam_IE_EAP()

    GetDeafult_Parameter(132);
    GetDeafult_Parameter(133);
    GetDeafult_Parameter(134);
    GetDeafult_Parameter(135);

END

FUNCTION DefaultToParam_FR_EAP()

```

```

        GetDeafult_Parameter(148);
        GetDeafult_Parameter(149);
        GetDeafult_Parameter(150);
        GetDeafult_Parameter(151);
        GetDeafult_Parameter(152);

END

FUNCTION DefaultToParam_MF_EAP()

    GetDeafult_Parameter(164);
    GetDeafult_Parameter(165);
    GetDeafult_Parameter(166);
    GetDeafult_Parameter(167);

END

FUNCTION DefaultToParam_VR_EAP()

    GetDeafult_Parameter(180);
    GetDeafult_Parameter(181);
    GetDeafult_Parameter(182);
    GetDeafult_Parameter(183);

END

FUNCTION DefaultToParam_CSB_SLC()

    GetDeafult_Parameter(212);
    GetDeafult_Parameter(213);
    GetDeafult_Parameter(214);

END

FUNCTION DefaultToParam_RVM_SLC()

    GetDeafult_Parameter(228);
    GetDeafult_Parameter(229);
    GetDeafult_Parameter(230);
    GetDeafult_Parameter(231);
    GetDeafult_Parameter(232);
    GetDeafult_Parameter(233);
    GetDeafult_Parameter(234);
    GetDeafult_Parameter(235);
    GetDeafult_Parameter(236);
    GetDeafult_Parameter(237);

END

FUNCTION DefaultToParam_PSM_SLC()

    GetDeafult_Parameter(244);
    GetDeafult_Parameter(245);
    GetDeafult_Parameter(246);
    GetDeafult_Parameter(247);
    GetDeafult_Parameter(248);
    GetDeafult_Parameter(249);

END

FUNCTION DefaultToParam_RRE_SLC()

    GetDeafult_Parameter(260);
    GetDeafult_Parameter(261);
    GetDeafult_Parameter(262);
    GetDeafult_Parameter(263);
    GetDeafult_Parameter(264);
    GetDeafult_Parameter(265);
    GetDeafult_Parameter(266);
    GetDeafult_Parameter(267);

END

FUNCTION DefaultToParam_PCM_SLC()

    GetDeafult_Parameter(276);
    GetDeafult_Parameter(277);
    GetDeafult_Parameter(278);
    GetDeafult_Parameter(279);
    GetDeafult_Parameter(280);
    GetDeafult_Parameter(281);
    GetDeafult_Parameter(282);

```

```
    GetDeafult_Parameter(283);  
END  
  
FUNCTION DefaultToParam_PBP_SLC()  
  
    GetDeafult_Parameter(292);  
    GetDeafult_Parameter(293);  
    GetDeafult_Parameter(294);  
    GetDeafult_Parameter(295);  
    GetDeafult_Parameter(296);  
    GetDeafult_Parameter(297);  
    GetDeafult_Parameter(298);  
  
END
```

```
//#####//  
//Functions used in Parametrization Screens in order to Save changes or Undo them.  
//#####//
```

```
FUNCTION Param_SaveAllChanges ()
```

```
    Update_Parameter();
```

```
    Update_Minimum();
```

```
    Update_Maximum();
```

```
    Update_Default();
```

```
    Update_Comand();
```

```
END
```

```
FUNCTION Param_UndoAllChanges ()
```

```
    Undo_Minimum();
```

```
    Undo_Maximum();
```

```
    Undo_Default();
```

```
    Undo_Parameter();
```

```
    Reset_StateComand();
```

```
END
```

```

//#####//

/* This file conatins functions that have to be exected constantly in the project.
   They are implemented in Cicode Obect Functions */

//#####//

//Function which is running persistently in all screens.
//When current hour is different from previous:
// * Services get new state value (enable/disable) according to current hour value from Operation Schedule
//   - Taking into consideration Comms protocol (MB/EAP).
// * Previous hour get its values from Scheduling screen to Operation, being shown in the begining of schedule
// * Power reference values get their values from Operation schedule current hour.
// * Power reference schedule from Operation screen is updated with the entry of values of 'current hour+23' from
Schedule Screen

GLOBAL INT Op_PreviousHour=0;    !Last hour worked with

FUNCTION Continuous_ScheduleActivation()

    INT i=0;
    INT hour;

    hour=TimeHour(TimeCurrent());    !Determine current system hour

    IF hour <> Op_PreviousHour THEN    !Comparing if current hour is different from previous

        //Services

        IF FLC_Comms=1 THEN !IF Modbus protocol

            Op_Schedule(aOp_IE[hour], 2, NRGLAB_Device\FLCcmdMB_IE); !IE get value from schedule array
            Op_Schedule(aOp_FR[hour], 3, NRGLAB_Device\FLCcmdMB_FR); !FR get value from schedule array
            Op_Schedule(aOp_MF[hour], 4, NRGLAB_Device\FLCcmdMB_MFA); !MFA get value from schedule array
            Op_Schedule(aOp_VR[hour], 5, NRGLAB_Device\FLCcmdMB_VR); !VR get value from schedule array

        ELSE !IF EAP protocol
            Op_Schedule(aOp_IE[hour], 8, NRGLAB_Device\FLCcmdEAP_IE); !IE get value from schedule array
            Op_Schedule(aOp_FR[hour], 9, NRGLAB_Device\FLCcmdEAP_FR); !FR get value from schedule array
            Op_Schedule(aOp_MF[hour], 10, NRGLAB_Device\FLCcmdEAP_MFA); !MFA get value from schedule array
            Op_Schedule(aOp_VR[hour], 11, NRGLAB_Device\FLCcmdEAP_VR); !VR get value from schedule array
        END

        Op_Schedule(aOp_RVM[hour], 14, NRGLAB_Device\SLCcmd_RVM); !RVM get value from schedule array
        Op_Schedule(aOp_PSM[hour], 15, NRGLAB_Device\SLCcmd_PSM); !PSM get value from schedule array
        Op_Schedule(aOp_RRE[hour], 16, NRGLAB_Device\SLCcmd_RRE); !RRE get value from schedule array
        Op_Schedule(aOp_PCM[hour], 17, NRGLAB_Device\SLCcmd_PCM); !PCM get value from schedule array
        Op_Schedule(aOp_PBP[hour], 18, NRGLAB_Device\SLCcmd_PBP); !PBP get value from schedule array

        //Powers

        NRGLAB_Device\FLCin_input_ActivePowerRef=Op_Pref[hour];    !Pref get value from schedule array
        NRGLAB_Device\FLCin_input_ReactivePowerRef=Op_Qref[hour];    !Qref get value from schedule array

        //Transfer service values from Scheduling aary to Opeartion array for hour=(Current hour)+23

        Scheduling aOp_IE[Op_ShowRollingHour(23)]=aSch_IE[Op_ShowRollingHour(23)];    !IE Op schedule gets value from
        Scheduling aOp_FR[Op_ShowRollingHour(23)]=aSch_FR[Op_ShowRollingHour(23)];    !FR Op schedule gets value from
        Scheduling aOp_MF[Op_ShowRollingHour(23)]=aSch_MF[Op_ShowRollingHour(23)];    !MFA Op schedule gets value from
        Scheduling aOp_VR[Op_ShowRollingHour(23)]=aSch_VR[Op_ShowRollingHour(23)];    !VR Op schedule gets value from
        Scheduling aOp_RVM[Op_ShowRollingHour(23)]=aSch_RVM[Op_ShowRollingHour(23)]; !RVM Op schedule gets value from
        Scheduling aOp_PSM[Op_ShowRollingHour(23)]=aSch_PSM[Op_ShowRollingHour(23)]; !PSM Op schedule gets value from
        Scheduling aOp_RRE[Op_ShowRollingHour(23)]=aSch_RRE[Op_ShowRollingHour(23)]; !RRE Op schedule gets value from
        Scheduling aOp_PCM[Op_ShowRollingHour(23)]=aSch_PCM[Op_ShowRollingHour(23)]; !PCM Op schedule gets value from
        Scheduling aOp_PBP[Op_ShowRollingHour(23)]=aSch_PBP[Op_ShowRollingHour(23)]; !PBP Op schedule gets value from

        //Transfer Power values from Scheduling array to Opeartion array for hour=(Current hour)+23

        Op_Pref[Op_ShowRollingHour(23)]=Sch_Pref[Op_ShowRollingHour(23)]; !Active Power Reference from

```



```
Scheduling to Operation
    Op_Qref[Op_ShowRollingHour(23)]=Sch_Qref[Op_ShowRollingHour(23)]; !Reactive Power Reference from
Scheduling to Operation

    //Previous hour = hour
    Op_PreviousHour=hour;

END

END
```

```

//#####//
//Functions with special use, can be used anytime and have an special behaveour.
//#####//

GLOBAL INT FirstCycle=1;

//#####//
//Function called in the first screen opened in the project and it is only called once.

FUNCTION Special_FirstCycle()

    IF FirstCycle=1 THEN

        loadMaximum();
        loadDefault();
        loadMinimum();
        loadComand();

        Transfer_Minimum();
        Transfer_Maximum();
        Transfer_Default();

        ForceCommandPCM();

        Sch_Save_ServiceSchedule();

        IF NRGLAB_Device\FLCin_statVar_Status <> 4 THEN
            FLC_Comms=1;
        END

    END

    FirstCycle=0;

END

//Function to Log in and Log out of user. Copied from PLC_Include and applied some modifications
MODULE INT mbLoginFormShown[25];

FUNCTION PLSLoginUser_NRGLab()

    //INT iPage = PageInfo(1);
    INT iPage = WinNumber();

    IF mbLoginFormShown[iPage] = TRUE THEN
        RETURN; //form already shown
    END

    //prevent multiple forms
    mbLoginFormShown[iPage] = TRUE;

    IF (UserInfo(0) <> "0") THEN
        // Confirm User Action
        IF (0 = Message(StrToLocalText("@(Confirm)"), StrToLocalText("@(Logout)"), 1+32)) THEN

            Logout();
            PageDisplay("Home");

        END
        mbLoginFormShown[iPage] = FALSE;
        RETURN;
    END

    LoginFormAdvanced();
    mbLoginFormShown[iPage] = FALSE;

    Sch_UndoService(); !Roger
    Param_UndoAllChanges(); !Roger

END

```

ANNEX 6

**CODI DE TWINCAT 3 PER LLEGIR DEL
CANAL MODBUS**

```
1  PROGRAM ModbusTest
2  VAR
3  ipAddress      : STRING ( 15 ) := '10.30.240.2' ; // Barcelona - Sevilla
4      nValue      : ARRAY [ 0 .. 2 ] OF INT ;
5
6      RValue_0     : ARRAY [ 0 .. 5 ] OF REAL ; //SCADA INPUTS
7      RValue_1     : ARRAY [ 0 .. 15 ] OF REAL ; //Monitoring Information
8      RValue_2     : ARRAY [ 0 .. 3 ] OF REAL ; //AS Measure
9      RValue_3     : ARRAY [ 0 .. 18 ] OF UINT ; //Commands
10     RValue_4     : ARRAY [ 0 .. 3 ] OF DINT ;
11     RValue_5     : ARRAY [ 0 .. 3 ] OF DINT ; //SCADA INPUTS Comand
12
13     fbReadRegs    : FB_MBReadRegs ;
14     fbReadRegs1   : FB_MBReadRegs ;
15     fbReadRegs2   : FB_MBReadRegs ;
16     fbReadRegs3   : FB_MBReadRegs ;
17     fbReadRegs4   : FB_MBReadRegs ;
18     fbReadRegs5   : FB_MBReadRegs ;
19     fbReadRegs6   : FB_MBReadRegs ;
20
21     SwapRW        : BOOL ;
22     mWritecounter : UDINT ;
23
24     mbreadcounter : UDINT ;
25     mbreadcounter1 : UDINT ;
26     mbreadcounter2 : UDINT ;
27     mbreadcounter3 : UDINT ;
28     mbreadcounter4 : UDINT ;
29
30     fbReadRegs0 : FB_MBReadRegs ;
31     mbreadcounter0 : UDINT ;
32     cont : UDINT ;
33     n_drand : drand ;
34     prueba : REAL ;
35     IntegralVal : REAL ;
36
37     sum : INT := 0 ;
38     contador : INT := 0 ;
39 END_VAR
40
```

```
1  SwapRW := NOT SwapRW ;
2  //Reading
3  // PCS Status
4      IF SwapRW AND ( NOT fbReadRegs6 . bBusy ) THEN
5          mbreadcounter := 0 ;
6          fbReadRegs6 . sIPAddr := ipAddress ;
7          fbReadRegs6 . nTCPPort := 502 ;
8          fbReadRegs6 . nQuantity := SIZEOF ( RValue_5 ) / 2 ;
9          fbReadRegs6 . nMBAAddr := 16#3274 ;
10         fbReadRegs6 . cbLength := SIZEOF ( RValue_5 ) ;
11         fbReadRegs6 . pDestAddr := ADR ( RValue_5 ) ;
12         fbReadRegs6 . tTimeout := T#10S ;
13         fbReadRegs6 ( bExecute := TRUE ) ;
14     ELSE
15         fbReadRegs6 ( bExecute := FALSE ) ;
```

```
16         IF fbReadRegs6 . bBusy THEN
17             mbreadcounter := mbreadcounter + 1 ;
18         END_IF
19     END_IF
20
21     Communications . ModbusToSCADA_Params . SCADA_Inputs . Commands := RValue_5
[ 0 ] ;
22
23     // PCS Status
24     IF SwapRW AND ( NOT fbReadRegs5 . bBusy ) THEN
25         mbreadcounter := 0 ;
26         fbReadRegs5 . sIPAddr := ipAddr ;
27         fbReadRegs5 . nTCPPort := 502 ;
28         fbReadRegs5 . nQuantity := SIZEOF ( RValue_4 ) / 2 ;
29         fbReadRegs5 . nMBAAddr := 16#3293 ;
30         fbReadRegs5 . cbLength := SIZEOF ( RValue_4 ) ;
31         fbReadRegs5 . pDestAddr := ADR ( RValue_4 ) ;
32         fbReadRegs5 . tTimeout := T#10S ;
33         fbReadRegs5 ( bExecute := TRUE ) ;
34     ELSE
35         fbReadRegs5 ( bExecute := FALSE ) ;
36         IF fbReadRegs5 . bBusy THEN
37             mbreadcounter := mbreadcounter + 1 ;
38         END_IF
39     END_IF
40
41     Communications . EtherCAT_Variables . Status := RValue_4 [ 0 ] ;
42
43     // Reading AS data from RTU
44     IF SwapRW AND ( NOT fbReadRegs . bBusy ) THEN
45         mbreadcounter := 0 ;
46         fbReadRegs . sIPAddr := ipAddr ;
47         fbReadRegs . nTCPPort := 502 ;
48         fbReadRegs . nQuantity := SIZEOF ( RValue_1 ) / 2 ;
49         fbReadRegs . nMBAAddr := 16#32A3 ;
50         fbReadRegs . cbLength := SIZEOF ( RValue_1 ) ;
51         fbReadRegs . pDestAddr := ADR ( RValue_1 ) ;
52         fbReadRegs . tTimeout := T#10S ;
53         fbReadRegs ( bExecute := TRUE ) ;
54     ELSE
55         fbReadRegs ( bExecute := FALSE ) ;
56         IF fbReadRegs . bBusy THEN
57             mbreadcounter := mbreadcounter + 1 ;
58         END_IF
59     END_IF
60
61     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables . BatSoc
:= RValue_1 [ 0 ] ;
62     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables . TotW
:= RValue_1 [ 7 ] ;
63     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables . TotVar
:= RValue_1 [ 8 ] ;
64     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables .
GridFre := RValue_1 [ 9 ] ;
65
66
```

```
67     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables . Vrmsa
:= RValue_1 [ 10 ] ;
68     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables . Vrmsb
:= RValue_1 [ 11 ] ;
69     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables . Vrmsc
:= RValue_1 [ 12 ] ;
70     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables . Irmsa
:= RValue_1 [ 13 ] ;
71     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables . Irmsb
:= RValue_1 [ 14 ] ;
72     Communications . ModbusToSCADA_Params . EtherCAT_Monitor_Variables . Irmsc
:= RValue_1 [ 15 ] ;
73
74
75     // Reading AOI data from RTU - first part
76     IF SwapRW AND ( NOT fbReadRegs1 . bBusy ) THEN
77         mbreadcounter1 := 0 ;
78         fbReadRegs1 . sIPAddr := ipAddr ;
79         fbReadRegs1 . nTCPPort := 502 ;
80         fbReadRegs1 . nQuantity := SIZEOF ( RValue_2 ) / 2 ;
81         fbReadRegs1 . nMBAAddr := 16#355C ;
82         fbReadRegs1 . cbLength := SIZEOF ( RValue_2 ) ;
83         fbReadRegs1 . pDestAddr := ADR ( RValue_2 ) ;
84         fbReadRegs1 . tTimeout := T#10S ;
85         fbReadRegs1 ( bExecute := TRUE ) ;
86     ELSE
87         fbReadRegs1 ( bExecute := FALSE ) ;
88         IF fbReadRegs1 . bBusy THEN
89             mbreadcounter1 := mbreadcounter1 + 1 ;
90         END_IF
91     END_IF
92
93     Communications . B_Athienou_11kV1_V := RValue_2 [ 0 ] ;
94
95     // Reading AOI data from RTU - second part
96     IF SwapRW AND ( NOT fbReadRegs2 . bBusy ) THEN
97         mbreadcounter2 := 0 ;
98         fbReadRegs2 . sIPAddr := ipAddr ;
99         fbReadRegs2 . nTCPPort := 502 ;
100        fbReadRegs2 . nQuantity := SIZEOF ( RValue_3 ) / 2 ;
101        fbReadRegs2 . nMBAAddr := 16#3000 ;
102        fbReadRegs2 . cbLength := SIZEOF ( RValue_3 ) ;
103        fbReadRegs2 . pDestAddr := ADR ( RValue_3 ) ;
104        fbReadRegs2 . tTimeout := T#10S ;
105        fbReadRegs2 ( bExecute := TRUE ) ;
106    ELSE
107        fbReadRegs2 ( bExecute := FALSE ) ;
108        IF fbReadRegs2 . bBusy THEN
109            mbreadcounter2 := mbreadcounter2 + 1 ;
110        END_IF
111    END_IF
112
113
114    Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
CommonBlock . Value := RValue_3 [ 0 ] ;
115    Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
```

```

116     InertiaEmulation . Value := RValue_3 [ 1 ] ;
117     Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
FrequencyRegulation . Value := RValue_3 [ 2 ] ;
118     Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
ModalFreq . Value := RValue_3 [ 3 ] ;
119     Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
VoltageRegulation . Value := RValue_3 [ 4 ] ;
120     Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
AdditionalService . Value := RValue_3 [ 5 ] ;
121
122     Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
CommonBlock_EAP . Value := RValue_3 [ 6 ] ;
123     Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
InertiaEmulation_EAP . Value := RValue_3 [ 7 ] ;
124     Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
FrequencyRegulation_EAP . Value := RValue_3 [ 8 ] ;
125     Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
ModalFreq_EAP . Value := RValue_3 [ 9 ] ;
126     Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
VoltageRegulation_EAP . Value := RValue_3 [ 10 ] ;
127     Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
Additonal_EAP . Value := RValue_3 [ 11 ] ;
128
129     Communications . ModbusToSCADA_Command . SecondLevel_Commands .
CommonComands . Value := RValue_3 [ 12 ] ;
130     Communications . ModbusToSCADA_Command . SecondLevel_Commands .
ResourceVariability . Value := RValue_3 [ 13 ] ;
131     Communications . ModbusToSCADA_Command . SecondLevel_Commands .
PowerShiftingManage . Value := RValue_3 [ 14 ] ;
132     Communications . ModbusToSCADA_Command . SecondLevel_Commands .
RampRateEnhancemet . Value := RValue_3 [ 15 ] ;
133     Communications . ModbusToSCADA_Command . SecondLevel_Commands .
PowerCurtaillment . Value := RValue_3 [ 16 ] ;
134     Communications . ModbusToSCADA_Command . SecondLevel_Commands .
PowerBalanceProv . Value := RValue_3 [ 17 ] ;
135     Communications . ModbusToSCADA_Command . SecondLevel_Commands .
AdditionalService . Value := RValue_3 [ 18 ] ;
136
137     // Reading AS data from RTU
138     IF SwapRW AND ( NOT fbReadRegs . bBusy ) THEN
139         mbreadcounter := 0 ;
140         fbReadRegs . sIPAddr := ipAddr ;
141         fbReadRegs . nTCPPort := 502 ;
142         fbReadRegs . nQuantity := SIZEOF ( RValue_0 ) / 2 ;
143         fbReadRegs . nMBAddr := 16#3275 ;
144         fbReadRegs . cbLength := SIZEOF ( RValue_0 ) ;
145         fbReadRegs . pDestAddr := ADR ( RValue_0 ) ;
146         fbReadRegs . tTimeout := T#10S ;
147         fbReadRegs ( bExecute := TRUE ) ;
148     ELSE
149         fbReadRegs ( bExecute := FALSE ) ;
150     IF fbReadRegs . bBusy THEN
151         mbreadcounter := mbreadcounter + 1 ;
152     END_IF
153 END_IF
154

```

```
155         Communications . ModbusToSCADA_Params . SCADA_Inputs .  
ActivePowerReference := RValue_0 [ 1 ] ;  
156         Communications . ModbusToSCADA_Params . SCADA_Inputs .  
ReactivePowerReference := RValue_0 [ 2 ] ;  
157  
158
```


ANNEX 7

**CODI DE TWINCAT 3 PER ENVIAR
DADES A IOT HUB UTILITZANT MQTT**

```

1  PROGRAM Azure_MQTT_Continuous
2  VAR
3
4      bInicializacion      :      BOOL      := TRUE ;
5      bConnectar           :      BOOL      := TRUE ;
6
7      //Cliente MQTT
8      fbMqttClient         :      FB_IoTMqttClient ;
9
10     //Mensaje JSON
11     sJson                 :      STRING ( 255 ) ;
12     fbCrearJson           :      FB_JsonSaxWriter ;
13
14     //Variables para publicar mensajes de dispositivos a nube
15     sTopicAlQuePublicar   :      STRING ( 255 ) ;
16     nMensajesEnviados     :      UDINT ;
17     fbTemporizador        :      TON ;
18     fbTemporizador2       :      TON ;
19
20     //Varibales para recibir mensajes de nube a dispositivo
21     bSubscrito            :      BOOL ;
22     sTopicSubscrito       :      STRING ( 255 ) ;
23     {attribute 'TcEncoding' := 'UTF-8'}
24     sTopicDatosRecibido   :      STRING ( 255 ) ;
25     {attribute 'TcEncoding' := 'UTF-8'}
26     sMensajeRecibido     :      STRING ( 255 ) ;
27     fbMessageQueue       :      FB_IotMqttMessageQueue ;
28     fbMessage             :      FB_IotMqttMessage ;
29
30     //Decodificar JSON
31     jsonDoc               :      SJsonValue ;
32     fbDecodificarJson     :      FB_JsonDomParser ;
33     jsonProp              :      SJsonValue ;
34
35     //Variables recibidas desde la nube
36     bOrdenesSalida1       :      BOOL ;
37     nOrdenAnalogica       :      LREAL ;
38
39     //Variables
40     CB_Bess               :      STRING ;
41     CB_11kV              :      STRING ;
42     sPCS                 :      STRING ;
43     Status               :      STRING ;
44     iPos                 :      INT ;
45     iValue               :      DINT ;
46     sComms               :      STRING ;
47
48 END_VAR
49
50 VAR CONSTANT
51
52     sDeviceId             :      STRING      := 'NRGLabSLC' ;
53     sRutaCertificadoAzure :      STRING ( 255 )      :=
54     'C:\Users\roger\OneDrive\Escritorio\Azure.cer' ;
55     sAzureSasToken        :      STRING ( 511 )      :=
56     'HostName=IoTHubNRGLab2020.azure-devices.net;DeviceId=NRGLabSLC;SharedAccessSi

```

```
sr=IoTHubNRGLab2020.azure-devices.net%2Fdevices%2FNRGLabSLC&sig=Dwi0RcJoxJAcH4/
;
55
56      Pi                      :   REAL                      :=      2 *
      3.1415926535897932384626433832795 ;

57  END_VAR
58
59
60

1  IF Communications . BESS_CB = 0 THEN
2      CB_Bess := 'Open' ;
3  ELSE
4      CB_Bess := 'Close' ;
5  END_IF
6
7  IF Communications . B_11kV_CB = 0 THEN
8      CB_11kV := 'Open' ;
9  ELSE
10     CB_11kV := 'Close' ;
11  END_IF
12
13
14  //Asignar un string a cada estado PCS
15  iValue := Communications . EtherCAT_Variables . Status ;
16  Azure . PCS_Stat . Status := iValue ;
17  sPCS := TO_STRING ( Azure . PCS_Stat . Status ) ;
18  iPos := FIND ( sPCS , '_' ) ;
19  Status := REPLACE ( sPCS , ' ' , 1 , iPos ) ;
20  iPos := FIND ( Status , '_' ) ;
21  IF iPos <> 0 THEN
22      Status := REPLACE ( Status , ' ' , 1 , iPos ) ;
23  END_IF
24
25
26  //Indicar tipo de canal de Comunicación: ModBus, EAP o Not Active
27  IF iValue < 2 OR iValue > 4 THEN
28      sComms := 'Not Active' ;
29  ELSIF iValue = 2 OR iValue = 3 THEN
30      sComms := 'ModBus' ;
31  ELSIF iValue = 4 THEN
32      sComms := 'EAP' ;
33  END_IF
34
35
36
37  IF bInicializacion THEN
38
39      bInicializacion          := FALSE ;
40
41      //Preparar topics a los que publicar y subscribirse
42
43      sTopicAlQuePublicar      :=
      'devices/NRGLabSLC/messages/events/readpipe/' ;
```

```
44      sTopicSubscrito          := 'devices/NRGLabSLC/messages/devicebound/#' ;
45
46      //Datos para la conexión a Azure IoT Hub por MQTT
47
48      fbMqttClient . stTLS . sCA      := sRutaCertificadoAzure ;
49      fbMqttClient . stTLS . sAzureSas := sAzureSasToken ;
50      fbMqttClient . ipMessageQueue   := fbMessageQueue ;
51
52      END_IF
53
54      fbMqttClient . Execute ( bConnectar ) ;
55
56      //Comprobar si se esta conectado
57      IF fbMqttClient . bConnected THEN
58
59          //Subscribirse al topic
60          IF NOT bSubscrito THEN
61              bSubscrito := fbMqttClient . Subscribe ( sTopic := sTopicSubscrito , eQos
:= TcIoTMqttQos . AtMostOnceDelivery ) ;
62          ELSE
63              //En el arranque se puede enviar un mensaje de inicializació, por
ejemplo la localizacion de la CPU o el estado
64          END_IF
65
66          //Dar la orden de envio de datos cada 60 segundo
67          fbTemporizador ( IN := TRUE , PT := T#60S ) ;
68
69          IF fbTemporizador . Q THEN
70
71              fbTemporizador ( IN := FALSE ) ;
72
73              //Mensaje 1: Señales de BESS: Soc, P, Q, freq
74
75              //Crear telegrama JSON a enviar
76              //Vaciar el objeto JSON
77              fbCrearJson . ResetDocument ( ) ;
78
79              //Añade la llave de apertura
80              fbCrearJson . StartObject ( ) ;
81              //Añade las parejas clave/valor al mensaje
82              fbCrearJson . AddKey ( 'SoC' ) ;
83              fbCrearJson . AddReal ( Communications . ModbusTOSCADa_Params .
EtherCAT_Monitor_Variables . BatSoc ) ;
84              fbCrearJson . AddKey ( 'Pconverter' ) ;
85              fbCrearJson . AddReal ( Communications . ModbusTOSCADa_Params .
EtherCAT_Monitor_Variables . TotW ) ;
86              fbCrearJson . AddKey ( 'Qconverter' ) ;
87              fbCrearJson . AddReal ( Communications . ModbusTOSCADa_Params .
EtherCAT_Monitor_Variables . TotVAr ) ;
88              fbCrearJson . AddKey ( 'Frequency' ) ;
89              fbCrearJson . AddReal ( Communications . ModbusToSCADA_Params .
EtherCAT_Monitor_Variables . GridFre / Pi ) ;
90
91              //Añade la llave de clausura
92              fbCrearJson . EndObject ( ) ;
93
```

```

94      //Obtiene el STRING con el mensaje JSON
95      sJson      := fbCrearJson . GetDocument ( ) ;
96
97      //Publicar el telegrama JSON a la cuenta de Azure
98      fbMqttClient . Publish ( sTopic := sTopicAlQuePublicar , pPayload := ADR
( sJson ) , nPayloadSize := TO_UDINT ( LEN ( sJson ) ) , eQoS := TcIotMqttQos .
AtMostOnceDelivery , bRetain := FALSE , bQueue := FALSE ) ;
99
100     //Llevar la cuenta de los mensajes publicados
101     nMensajesEnvidados      := nMensajesEnvidados + 1 ;
102
103
104     //Mensaje 2: Señales de de Tensión e Intensidad de las tres fases
(a,b,c) y Tensión en bus de 11kV
105
106     fbCrearJson . ResetDocument ( ) ;
107     //fbCrearJson();
108     //Añade la llave de apertura
109     fbCrearJson . StartObject ( ) ;
110
111     fbCrearJson . AddKey ( 'Va_LV' ) ;
112     fbCrearJson . AddReal ( Communications . ModbusTOSCADA_Params .
EtherCAT_Monitor_Variables . Vrmsa ) ;
113     fbCrearJson . AddKey ( 'Vb_LV' ) ;
114     fbCrearJson . AddReal ( Communications . ModbusTOSCADA_Params .
EtherCAT_Monitor_Variables . Vrmsb ) ;
115     fbCrearJson . AddKey ( 'Vc_LV' ) ;
116     fbCrearJson . AddReal ( Communications . ModbusTOSCADA_Params .
EtherCAT_Monitor_Variables . Vrmsc ) ;
117     fbCrearJson . AddKey ( 'Ia_LV' ) ;
118     fbCrearJson . AddReal ( Communications . ModbusTOSCADA_Params .
EtherCAT_Monitor_Variables . Irmsa ) ;
119     fbCrearJson . AddKey ( 'Ib_LV' ) ;
120     fbCrearJson . AddReal ( Communications . ModbusTOSCADA_Params .
EtherCAT_Monitor_Variables . Irmsb ) ;
121     fbCrearJson . AddKey ( 'Ic_LV' ) ;
122     fbCrearJson . AddReal ( Communications . ModbusTOSCADA_Params .
EtherCAT_Monitor_Variables . Irmsc ) ;
123     fbCrearJson . AddKey ( 'V_11kV' ) ;
124     fbCrearJson . AddReal ( Communications . B_Athienou_11kV1_V ) ;
125
126     fbCrearJson . EndObject ( ) ;
127
128     //Obtiene el STRING con el mensaje JSON
129     sJson      := fbCrearJson . GetDocument ( ) ;
130
131     //Publicar el telegrama JSON a la cuenta de Azure
132     fbMqttClient . Publish ( sTopic := sTopicAlQuePublicar , pPayload := ADR
( sJson ) , nPayloadSize := TO_UDINT ( LEN ( sJson ) ) , eQoS := TcIotMqttQos .
AtMostOnceDelivery , bRetain := FALSE , bQueue := FALSE ) ;
133
134     //Llevar la cuenta de los mensajes publicados
135     nMensajesEnvidados      := nMensajesEnvidados + 1 ;
136
137     //Mensaje 3: Estados de los disyuntores, estado de la máquina de
estados, protocolo de coms. en FLC actualmente y potencias de referencia en

```

```

138
139         fbCrearJson . ResetDocument ( ) ;
140         fbCrearJson . StartObject ( ) ;
141
142         fbCrearJson . AddKey ( 'CB_Bess' ) ;
143         fbCrearJson . AddString ( CB_Bess ) ;
144         fbCrearJson . AddKey ( 'CB_11kV' ) ;
145         fbCrearJson . AddString ( CB_11kV ) ;
146         fbCrearJson . AddKey ( 'PCS_Status' ) ;
147         fbCrearJson . AddString ( Status ) ;
148         fbCrearJson . AddKey ( 'CommsProtocol' ) ;
149         fbCrearJson . AddString ( sComms ) ;
150         fbCrearJson . AddKey ( 'Pref' ) ;
151         fbCrearJson . AddReal ( Communications . ModbusToSCADA_Params .
SCADA_Inputs . ActivePowerReference * 1000 ) ;
152         fbCrearJson . AddKey ( 'Qref' ) ;
153         fbCrearJson . AddReal ( Communications . ModbusToSCADA_Params .
SCADA_Inputs . ReactivePowerReference * 1000 ) ;
154
155         fbCrearJson . EndObject ( ) ;
156         //Obtiene el STRING con el mensaje JSON
157         sJson := fbCrearJson . GetDocument ( ) ;
158         //Publicar el telegrama JSON a la cuenta de Azure
159         fbMqttClient . Publish ( sTopic := sTopicAlQuePublicar , pPayload := ADR
( sJson ) , nPayloadSize := TO_UDINT ( LEN ( sJson ) ) , eQoS := TcIotMqttQos .
AtMostOnceDelivery , bRetain := FALSE , bQueue := FALSE ) ;
160
161
162         //Mensaje 2: Estado de los servicios de FLC MB, FLC EAP y SLC
163
164         fbCrearJson . ResetDocument ( ) ;
165         fbCrearJson . StartObject ( ) ;
166
167         fbCrearJson . AddKey ( 'IE_MB' ) ;
168         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_Modbus . InertiaEmulation . Bits . Enable_Inertia ) ;
169         fbCrearJson . AddKey ( 'FR_MB' ) ;
170         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_Modbus . FrequencyRegulation . Bits . Enable_DroopP ) ;
171         fbCrearJson . AddKey ( 'MFA_MB' ) ;
172         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_Modbus . ModalFreq . Bits . Enable_ModalFreq ) ;
173         fbCrearJson . AddKey ( 'VR_MB' ) ;
174         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_Modbus . VoltageRegulation . Bits . Enable_VoltageRef ) ;
175         fbCrearJson . AddKey ( 'IE_EAP' ) ;
176         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_EAP . InertiaEmulation_EAP . Bits . Enable_Inertia ) ;
177         fbCrearJson . AddKey ( 'FR_EAP' ) ;
178         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_EAP . FrequencyRegulation_EAP . Bits . Enable_DroopP ) ;
179         fbCrearJson . AddKey ( 'MFA_EAP' ) ;
180         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_EAP . ModalFreq_EAP . Bits . Enable_ModalFreq ) ;
181         fbCrearJson . AddKey ( 'VR_EAP' ) ;
182         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .

```

```

EtherCAT_Commands_EAP . VoltageRegulation_EAP . Bits . Enable_VoltageRegulation
) ;
183         fbCrearJson . AddKey ( 'RVM' ) ;
184         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . ResourceVariability . Bits . Enable_RVM ) ;
185         fbCrearJson . AddKey ( 'PSM' ) ;
186         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . PowerShiftingManage . Bits . Enable_PSM ) ;
187         fbCrearJson . AddKey ( 'RRE' ) ;
188         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . RampRateEnhancemet . Bits . Enable_RRE ) ;
189         fbCrearJson . AddKey ( 'PCM' ) ;
190         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . PowerCurtailment . Bits . Enable_PCM ) ;
191         fbCrearJson . AddKey ( 'PBP' ) ;
192         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . PowerBalanceProv . Bits . Enable_PBP ) ;
193
194         fbCrearJson . EndObject ( ) ;
195         //Obtiene el STRING con el mensaje JSON
196         sJson := fbCrearJson . GetDocument ( ) ;
197         //Publicar el telegrama JSON a la cuenta de Azure
198         fbMqttClient . Publish ( sTopic := sTopicAlQuePublicar , pPayload := ADR
( sJson ) , nPayloadSize := TO_UDINT ( LEN ( sJson ) ) , eQoS := TcIotMqttQos .
AtMostOnceDelivery , bRetain := FALSE , bQueue := FALSE ) ;
199
200         END_IF
201
202         //Si cambia algun valor en el mensaje 3 antes de que se reinicie el
temporizador se envía mensaje de nuevo
203
204         IF ( CB_Bess <> Azure . CB_BessPrevious ) OR ( CB_11kV <> Azure .
CB_11kVPrevious ) OR ( Communications . EtherCAT_Variables . Status <> Azure .
OpStatusPrevious ) OR ( Communications . ModbusToSCADA_Params . SCADA_Inputs .
ActivePowerReference <> Azure . Pref_Previous ) OR ( Communications .
ModbusToSCADA_Params . SCADA_Inputs . ReactivePowerReference <> Azure .
Qref_Previous ) THEN
205
206             fbCrearJson . ResetDocument ( ) ;
207             fbCrearJson . StartObject ( ) ;
208
209             fbCrearJson . AddKey ( 'CB_Bess' ) ;
210             fbCrearJson . AddString ( CB_Bess ) ;
211             fbCrearJson . AddKey ( 'CB_11kV' ) ;
212             fbCrearJson . AddString ( CB_11kV ) ;
213             fbCrearJson . AddKey ( 'PCS_Status' ) ;
214             fbCrearJson . AddString ( Status ) ;
215             fbCrearJson . AddKey ( 'CommsProtocol' ) ;
216             fbCrearJson . AddString ( sComms ) ;
217             fbCrearJson . AddKey ( 'Pref' ) ;
218             fbCrearJson . AddReal ( Communications . ModbusToSCADA_Params .
SCADA_Inputs . ActivePowerReference * 1000 ) ;
219             fbCrearJson . AddKey ( 'Qref' ) ;
220             fbCrearJson . AddReal ( Communications . ModbusToSCADA_Params .
SCADA_Inputs . ReactivePowerReference * 1000 ) ;
221
222             fbCrearJson . EndObject ( ) ;

```

```

223      //Obtiene el STRING con el mensaje JSON
224      sJson := fbCrearJson . GetDocument ( ) ;
225      //Publicar el telegrama JSON a la cuenta de Azure
226      fbMqttClient . Publish ( sTopic := sTopicAlQuePublicar , pPayload := ADR
( sJson ) , nPayloadSize := TO_UDINT ( LEN ( sJson ) ) , eQoS := TcIotMqttQos .
AtMostOnceDelivery , bRetain := FALSE , bQueue := FALSE ) ;

227
228      //Se guarda el último valor enviado para comparar si suceden cambios
229
230      Azure . CB_BessPrevious := CB_Bess ;
231      Azure . CB_11kVPrevious := CB_11kV ;
232      Azure . OpStatusPrevious := Communications . EtherCAT_Variables .
Status ;
233      Azure . Pref_Previous := Communications . ModbusToSCADA_Params .
SCADA_Inputs . ActivePowerReference ;
234      Azure . Qref_Previous := Communications . ModbusToSCADA_Params .
SCADA_Inputs . ReactivePowerReference ;
235
236      END_IF
237
238      //Si cambia el valor de algun servicio antes de que el temporizador se
reinicie se envia el mensaje 4
239
240      IF ( Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
InertiaEmulation . Bits . Enable_Inertia <> Azure . OperationPreviousSchedule [
1 ] ) OR
241      ( Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
FrequencyRegulation . Bits . Enable_DroopP <> Azure .
OperationPreviousSchedule [ 2 ] ) OR
242      ( Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
ModalFreq . Bits . Enable_ModalFreq <> Azure . OperationPreviousSchedule [ 3 ] )
OR
243      ( Communications . ModbusToSCADA_Command . EtherCAT_Commands_Modbus .
VoltageRegulation . Bits . Enable_VoltageRef <> Azure .
OperationPreviousSchedule [ 4 ] ) OR
244      ( Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
InertiaEmulation_EAP . Bits . Enable_Inertia <> Azure .
OperationPreviousSchedule [ 5 ] ) OR
245      ( Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
FrequencyRegulation_EAP . Bits . Enable_DroopP <> Azure .
OperationPreviousSchedule [ 6 ] ) OR
246      ( Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
ModalFreq_EAP . Bits . Enable_ModalFreq <> Azure . OperationPreviousSchedule [
7 ] ) OR
247      ( Communications . ModbusToSCADA_Command . EtherCAT_Commands_EAP .
VoltageRegulation_EAP . Bits . Enable_VoltageRegulation <> Azure .
OperationPreviousSchedule [ 8 ] ) OR
248      ( Communications . ModbusToSCADA_Command . SecondLevel_Commands .
ResourceVariability . Bits . Enable_RVM <> Azure . OperationPreviousSchedule [
9 ] ) OR
249      ( Communications . ModbusToSCADA_Command . SecondLevel_Commands .
PowerShiftingManage . Bits . Enable_PSM <> Azure . OperationPreviousSchedule [
10 ] ) OR
250      ( Communications . ModbusToSCADA_Command . SecondLevel_Commands .
RampRateEnhancemet . Bits . Enable_RRE <> Azure . OperationPreviousSchedule [
11 ] ) OR

```



```

251         ( Communications . ModbusToSCADA_Command . SecondLevel_Commands .
PowerCurtailment . Bits . Enable_PCM <> Azure . OperationPreviousSchedule [ 12 ]
) OR
252         ( Communications . ModbusToSCADA_Command . SecondLevel_Commands .
PowerBalanceProv . Bits . Enable_PBP <> Azure . OperationPreviousSchedule [ 13 ]
) THEN
253
254
255         fbCrearJson . ResetDocument ( ) ;
256         fbCrearJson . StartObject ( ) ;
257
258         fbCrearJson . AddKey ( 'IE_MB' ) ;
259         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_Modbus . InertiaEmulation . Bits . Enable_Inertia ) ;
260         fbCrearJson . AddKey ( 'FR_MB' ) ;
261         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_Modbus . FrequencyRegulation . Bits . Enable_DroopP ) ;
262         fbCrearJson . AddKey ( 'MFA_MB' ) ;
263         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_Modbus . ModalFreq . Bits . Enable_ModalFreq ) ;
264         fbCrearJson . AddKey ( 'VR_MB' ) ;
265         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_Modbus . VoltageRegulation . Bits . Enable_VoltageRef ) ;
266         fbCrearJson . AddKey ( 'IE_EAP' ) ;
267         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_EAP . InertiaEmulation_EAP . Bits . Enable_Inertia ) ;
268         fbCrearJson . AddKey ( 'FR_EAP' ) ;
269         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_EAP . FrequencyRegulation_EAP . Bits . Enable_DroopP ) ;
270         fbCrearJson . AddKey ( 'MFA_EAP' ) ;
271         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_EAP . ModalFreq_EAP . Bits . Enable_ModalFreq ) ;
272         fbCrearJson . AddKey ( 'VR_EAP' ) ;
273         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
EtherCAT_Commands_EAP . VoltageRegulation_EAP . Bits . Enable_VoltageRegulation
) ;
274         fbCrearJson . AddKey ( 'RVM' ) ;
275         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . ResourceVariability . Bits . Enable_RVM ) ;
276         fbCrearJson . AddKey ( 'PSM' ) ;
277         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . PowerShiftingManage . Bits . Enable_PSM ) ;
278         fbCrearJson . AddKey ( 'RRE' ) ;
279         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . RampRateEnhancemet . Bits . Enable_RRE ) ;
280         fbCrearJson . AddKey ( 'PCM' ) ;
281         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . PowerCurtailment . Bits . Enable_PCM ) ;
282         fbCrearJson . AddKey ( 'PBP' ) ;
283         fbCrearJson . AddBool ( Communications . ModbusToSCADA_Command .
SecondLevel_Commands . PowerBalanceProv . Bits . Enable_PBP ) ;
284
285         fbCrearJson . EndObject ( ) ;
286         //Obtiene el STRING con el mensaje JSON
287         sJson := fbCrearJson . GetDocument ( ) ;
288         //Publicar el telegrama JSON a la cuenta de Azure

```

```

289         fbMqttClient . Publish ( sTopic := sTopicAlQuePublicar , pPayload := ADR
        ( sJson ) , nPayloadSize := TO_UDINT ( LEN ( sJson ) ) , eQoS := TcIotMqttQos .
        AtMostOnceDelivery , bRetain := FALSE , bQueue := FALSE ) ;

290
291         //Se guarda el último valor enviado para comparar si suceden cambios
292
293         Azure . OperationPreviousSchedule [ 1 ] := Communications .
        ModbusToSCADA_Command . EtherCAT_Commands_Modbus . InertiaEmulation . Bits .
        Enable_Inertia ;
294         Azure . OperationPreviousSchedule [ 2 ] := Communications .
        ModbusToSCADA_Command . EtherCAT_Commands_Modbus . FrequencyRegulation . Bits .
        Enable_DroopP ;
295         Azure . OperationPreviousSchedule [ 3 ] := Communications .
        ModbusToSCADA_Command . EtherCAT_Commands_Modbus . ModalFreq . Bits .
        Enable_ModalFreq ;
296         Azure . OperationPreviousSchedule [ 4 ] := Communications .
        ModbusToSCADA_Command . EtherCAT_Commands_Modbus . VoltageRegulation . Bits .
        Enable_VoltageRef ;
297         Azure . OperationPreviousSchedule [ 5 ] := Communications .
        ModbusToSCADA_Command . EtherCAT_Commands_EAP . InertiaEmulation_EAP . Bits .
        Enable_Inertia ;
298         Azure . OperationPreviousSchedule [ 6 ] := Communications .
        ModbusToSCADA_Command . EtherCAT_Commands_EAP . FrequencyRegulation_EAP . Bits .
        Enable_DroopP ;
299         Azure . OperationPreviousSchedule [ 7 ] := Communications .
        ModbusToSCADA_Command . EtherCAT_Commands_EAP . ModalFreq_EAP . Bits .
        Enable_ModalFreq ;
300         Azure . OperationPreviousSchedule [ 8 ] := Communications .
        ModbusToSCADA_Command . EtherCAT_Commands_EAP . VoltageRegulation_EAP . Bits .
        Enable_VoltageRegulation ;
301         Azure . OperationPreviousSchedule [ 9 ] := Communications .
        ModbusToSCADA_Command . SecondLevel_Commands . ResourceVariability . Bits .
        Enable_RVM ;
302         Azure . OperationPreviousSchedule [ 10 ] := Communications .
        ModbusToSCADA_Command . SecondLevel_Commands . PowerShiftingManage . Bits .
        Enable_PSM ;
303         Azure . OperationPreviousSchedule [ 11 ] := Communications .
        ModbusToSCADA_Command . SecondLevel_Commands . RampRateEnhancemet . Bits .
        Enable_RRE ;
304         Azure . OperationPreviousSchedule [ 12 ] := Communications .
        ModbusToSCADA_Command . SecondLevel_Commands . PowerCurtailement . Bits .
        Enable_PCM ;
305         Azure . OperationPreviousSchedule [ 13 ] := Communications .
        ModbusToSCADA_Command . SecondLevel_Commands . PowerBalanceProv . Bits .
        Enable_PBP ;
306
307         END_IF
308
309
310     END_IF
311

```

ANNEX 8

MANUAL DE CONNEXIÓ I ENVIAMENT DE DADES A IOT HUB DE BECKHOFF

Comunicación entre TwinCAT y Microsoft Azure IoT Hub

TABLA DE CONTENIDOS

| | | |
|-----------|---|-----------|
| 1. | OBJETO DEL DOCUMENTO | 2 |
| 2. | CONFIGURACIÓN DEL AZURE IOT HUB | 3 |
| 3. | COMUNICAR CON EL TF6701 IOT COMMUNICATION | 9 |
| | Generar el SAS Token | 9 |
| | Generar el certificado | 10 |
| | Programa de PLC | 13 |
| | <i>Conexión con el dispositivo de Azure IoT Hub en TwinCAT</i> | 13 |
| | <i>Publicar datos del dispositivo a la nube en TwinCAT</i> | 14 |
| | <i>Lectura de mensajes de nube a dispositivo en TwinCAT</i> | 16 |
| | <i>Código de TwinCAT completo</i> | 18 |
| 4. | COMUNICAR CON EL TF6720 TC3 IOT DATA AGENT | 21 |
| 5. | GUARDAR DATOS EN AZURE SQL | 33 |
| | Configurar la base de datos de SQL en Azure | 33 |
| | Configurar el Stream Analytics Job | 38 |
| | Visualizar los datos guardados en la base de datos desde el portal de Azure | 45 |
| | Visualizar los datos guardados en la base de datos desde Visual Studio | 45 |
| 6. | VISUALIZACIÓN EN POWER BI | 50 |
| | Configurar el Stream Analytics Job | 50 |
| | Configuración del panel de control en Power bi | 58 |

1. OBJETO DEL DOCUMENTO

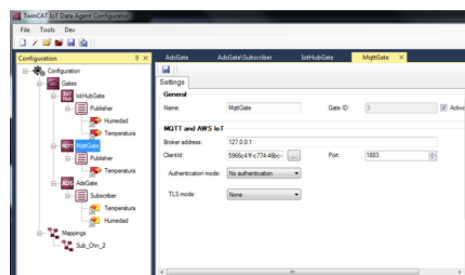
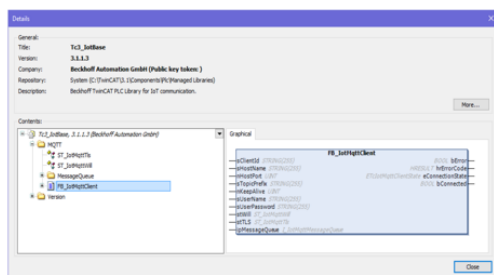
En este documento se explican los pasos necesarios para comunicar una CPU con TwinCAT con un dispositivo IoT Hub de Microsoft Azure para poder enviar fácilmente datos de máquina.

En el [segundo capítulo](#) de esta guía se explica la configuración básica necesaria en Microsoft Azure, en concreto como crear el dispositivo del IoT Hub con el que se comunicará y como generar sus datos de acceso¹.

En los capítulos tercero y cuarto se explica la programación y configuración básica en el entorno de TwinCAT para realizar esta comunicación. En el [capítulo 3](#) se hace uso del suplemento 'TF6701 TC3 IoT Communication' que proporciona unas librerías comunicar con la nube desde código de PLC. En el [capítulo 4](#) se hace uso del suplemento 'TF6720 TC3 IoT Data Agent' que proporciona un asistente grafico para poder configurar las variables que se van a subir sin necesidad de escribir código.

En la siguiente imagen se pueden ver las principales diferencias entre ambos suplementos para la subida de datos:

- | | |
|---|---|
| <ul style="list-style-type: none"> • TF6701 TC3 IoT Communication <ul style="list-style-type: none"> • Librería para TC3 PLC. • Recogida de datos en Tiempo Real. • Posibilidad de compartir cualquier dato que llegue a TwinCAT (PLC, buses de campo, Simulink, etc...). • Posibilidad de comunicar simultáneamente con tantos brokers como se desee. | <ul style="list-style-type: none"> • TF6720 TC3 IoT Data Agent <ul style="list-style-type: none"> • Configuración mediante un asistente (sin necesidad de TwinCAT PLC). • No recoge los datos en Tiempo Real. • Posibilidad de compartir cualquier dato que llegue por OPC UA o ADS. • Posibilidad de comunicar simultáneamente con tantos brokers como se desee. • La licencia depende del número de elementos OPC UA y ADS de los que se recojan datos. |
|---|---|

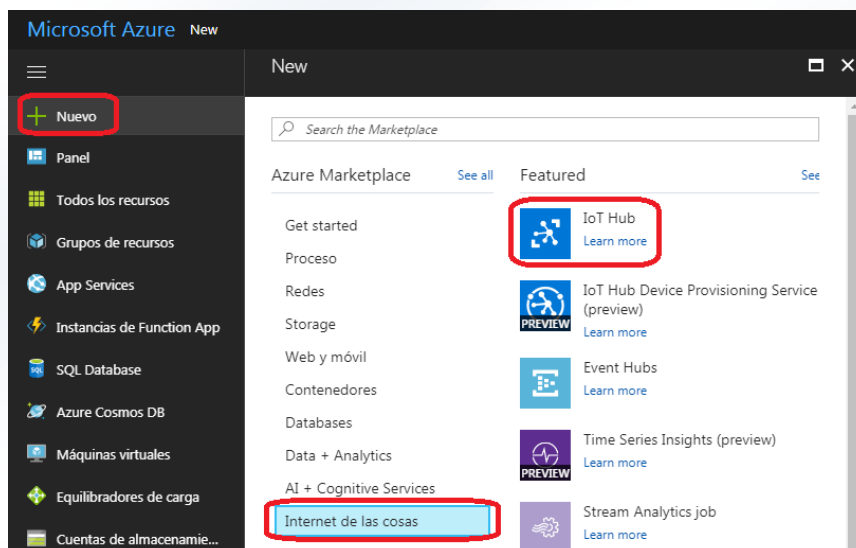


Los capítulos quinto y sexto muestran dos maneras de visualizar los datos que se han subido al dispositivo IoT Hub de Azure. En el [capítulo 5](#) se explica cómo guardar estos datos en una base de datos SQL en la propia suite de Azure, por ejemplo, para un análisis posterior. En el [capítulo 6](#) se explica el uso de la herramienta de Microsoft Power Bi para poder hacer un panel de control que muestre los datos subidos en tiempo real.

¹ Los dispositivos de Microsoft Azure IoT Hub utilizados en esta guía actualmente no existen. Se han dejado sus datos de acceso intencionadamente para poder compararlos con los que salgan en el momento de realizar la aplicación.

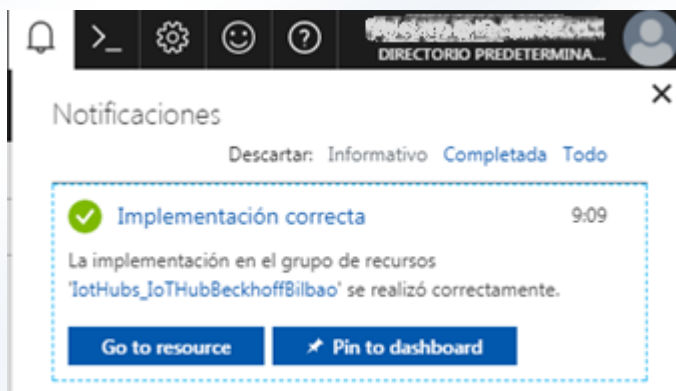
2. CONFIGURACIÓN DEL AZURE IOT HUB

En primer lugar, es necesario crear el centro de Azure IoT Hub con el que se comunicará desde TwinCAT. Para ello es necesario ir al portal de la suite de Microsoft Azure ([enlace](#)) y seleccionar la siguiente opción:

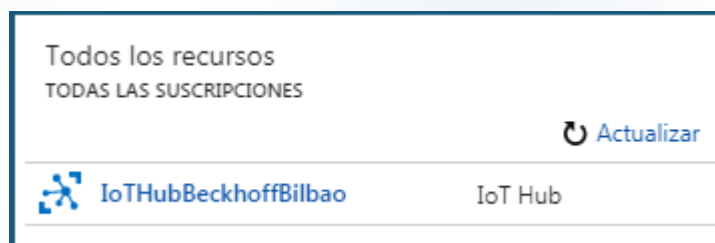


Entre los datos del IoT Hub es necesario seleccionar un nombre (debe ser un nombre único), el tamaño que tendrá (lo que afectará al precio), la suscripción y el grupo de recursos donde se hospedará. En caso de no tener ninguno creado se puede seleccionar la opción de crear uno nuevo:

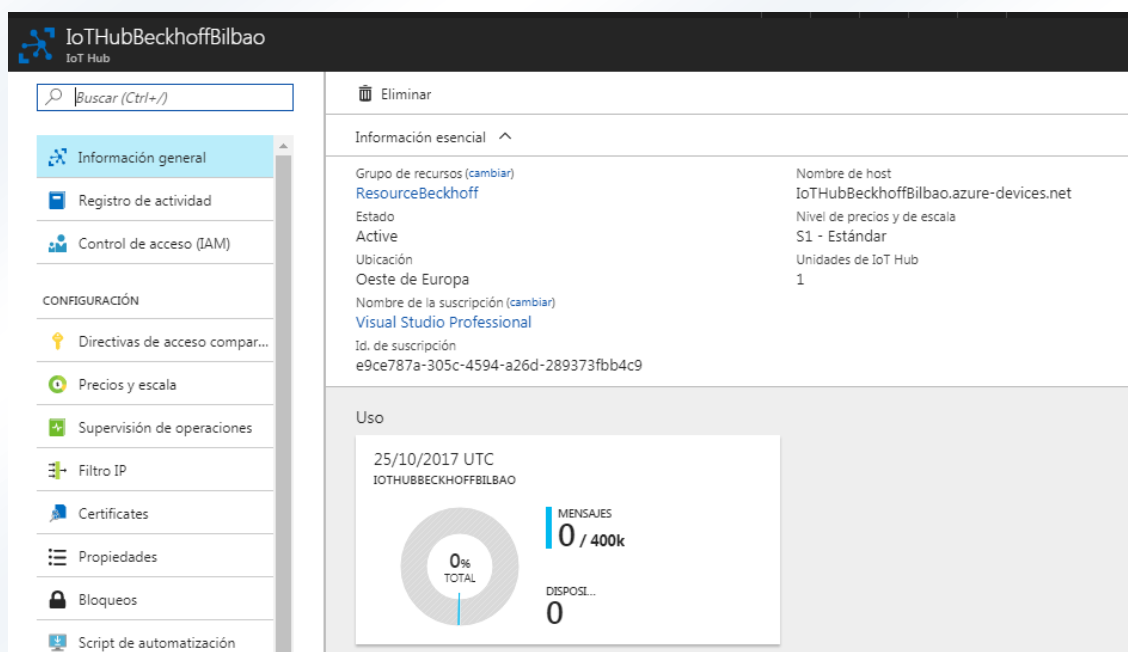
Una vez se le dé al botón de 'Crear' habrá que esperar a que aparezca la notificación de que se ha creado correctamente (puede tardar unos minutos):



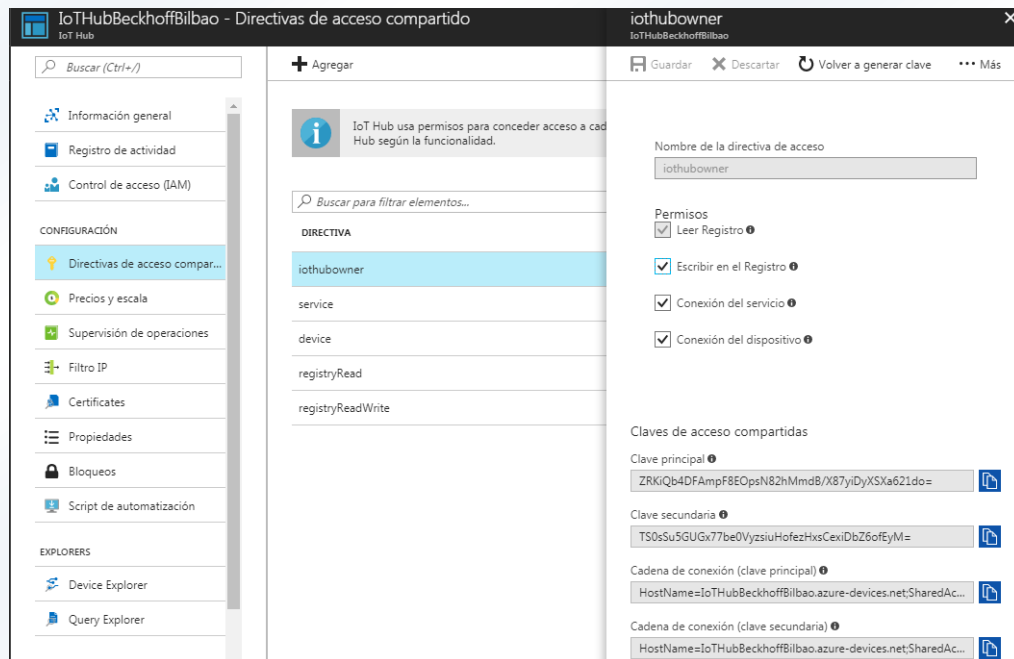
Desde el panel de control se puede acceder al centro IoT Hub creado pinchando sobre el:



Si se pincha sobre él se accederá al administrador de este centro IoT Hub, entre otras cosas, en la sección de 'Información general' se podrá ver la dirección completa, el número de mensajes consumidos (el máximo dependerá del tamaño seleccionado cuando se creó el dispositivo):



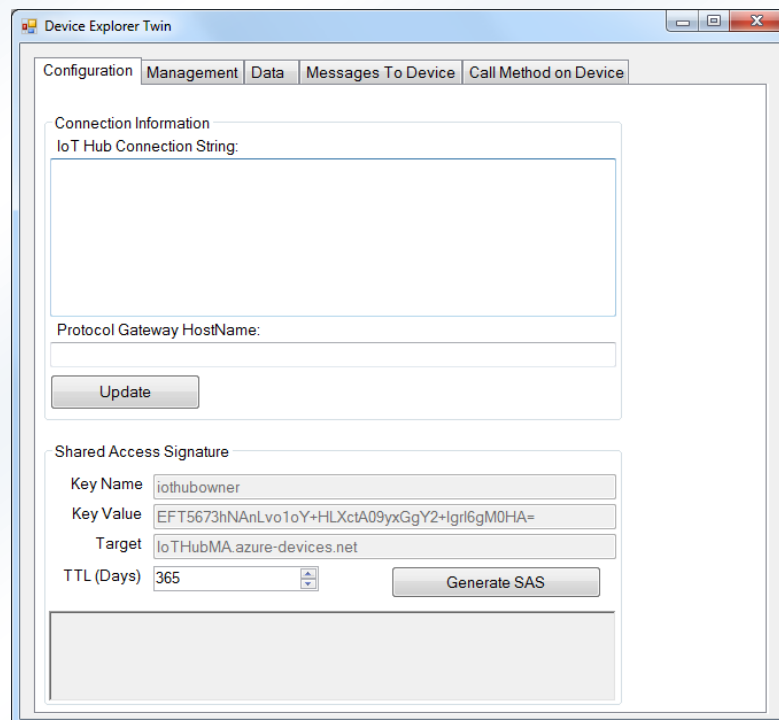
Bajo la sección de 'Directivas de acceso compartido' se pueden acceder a las claves de acceso de este IoT Hub:



El siguiente paso es crear un dispositivo sobre este IoT Hub. Cada CPU que se desee comunicar con la nube de Azure deberá atacar a un dispositivo diferente².

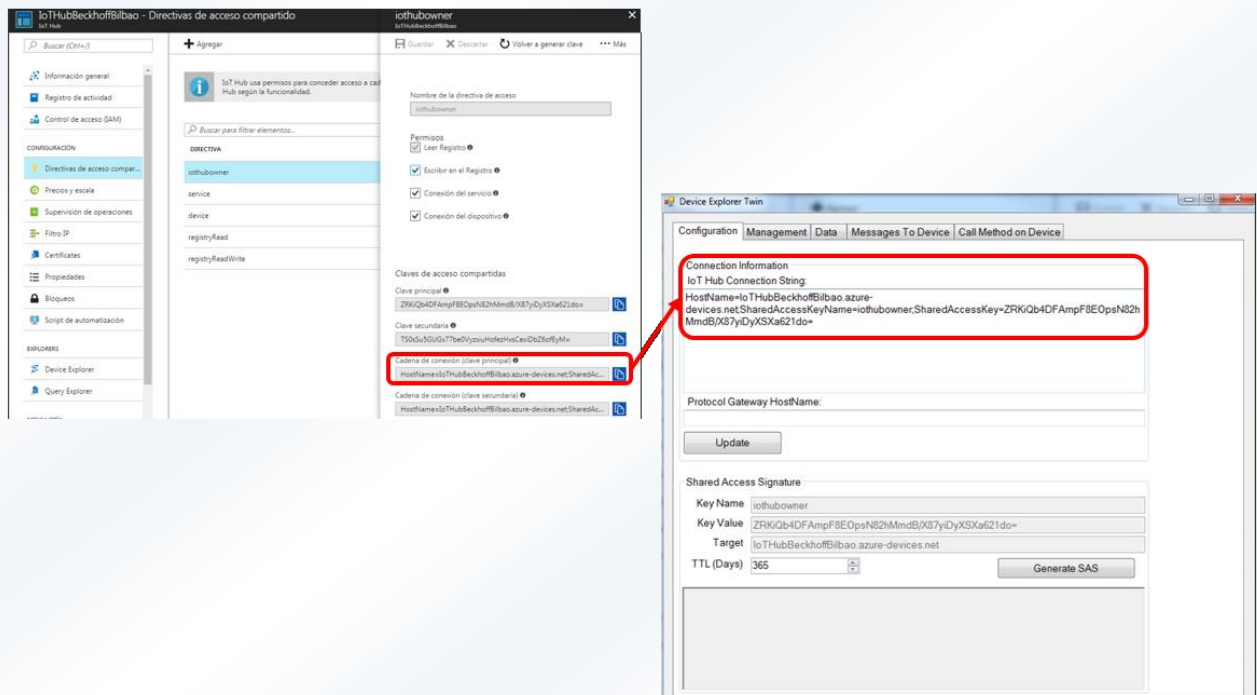
El centro de IoT Hub se puede gestionar desde aplicaciones desarrolladas en .NET ([enlace de información](#)), desde el CLI del portal de Azure ([enlace de información](#)) o, de una forma más sencilla, utilizando la aplicación del Device Explorer desarrollada por Microsoft ([enlace de descarga](#)). En esta guía se utilizará esta aplicación.

Una vez abierta la aplicación del Device Explorer, en la primera pantalla es necesario añadir la cadena de conexión principal con el Azure IoT Hub.



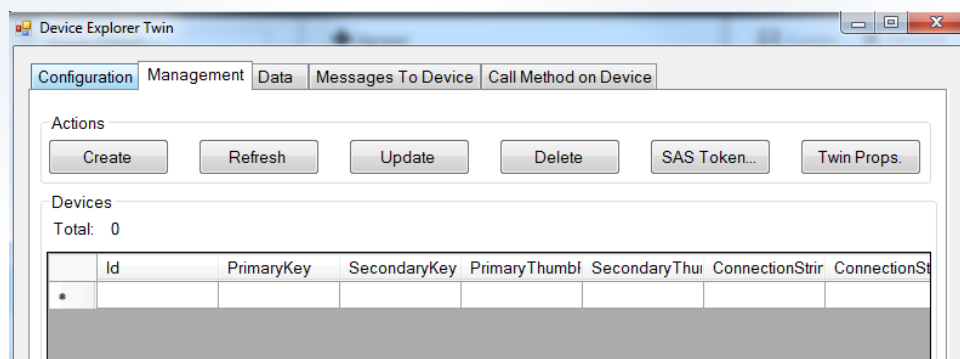
² Con el Data Agent se pueden comunicar varias CPUs a un único dispositivo ya que comunica por AMQP, con el IoT Communication cada dispositivo solo puede ser accedido desde una única CPU debido al SAS Token necesario para la comunicación MQTT.

Esta clave de conexión se obtiene del portal de Azure, en la sección de 'Directivas de acceso compartido':

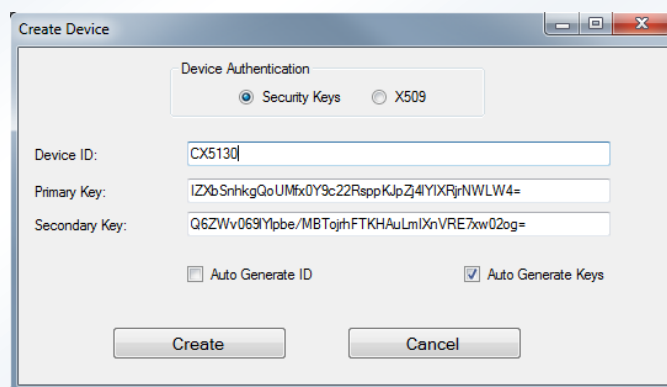


Una vez copiada la cadena de conexión es necesario darle al botón de 'Update' y ya se pueden crear dispositivos, monitorizar los datos, enviar mensajes de nube a dispositivo, etc...

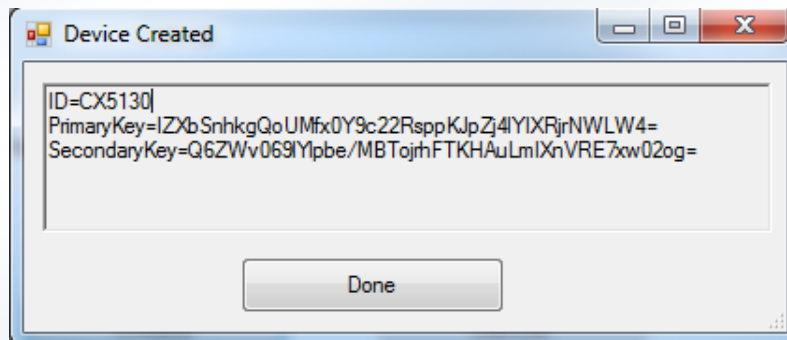
Para crear el dispositivo hay que ir a la pestaña de 'Management':



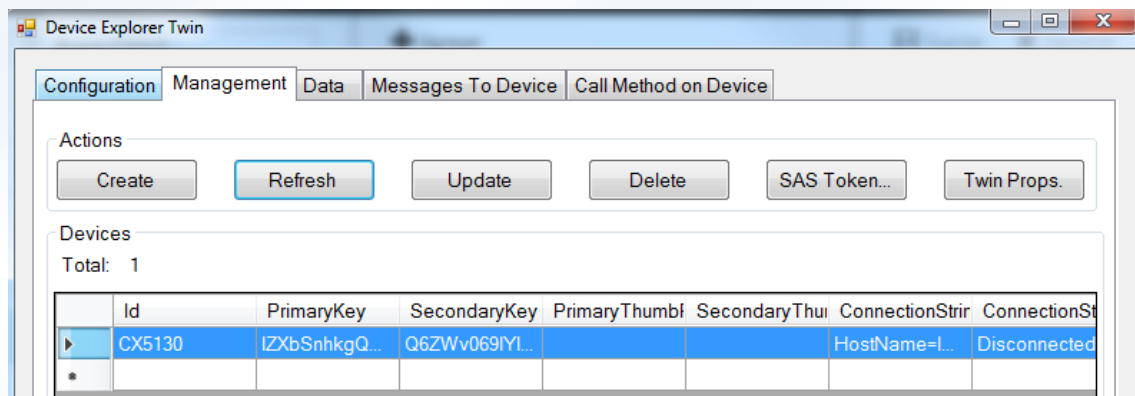
En esta pestaña hay que pulsar el botón de 'Create' y en el asistente que aparecerá darle un nombre al dispositivo que se va a crear:



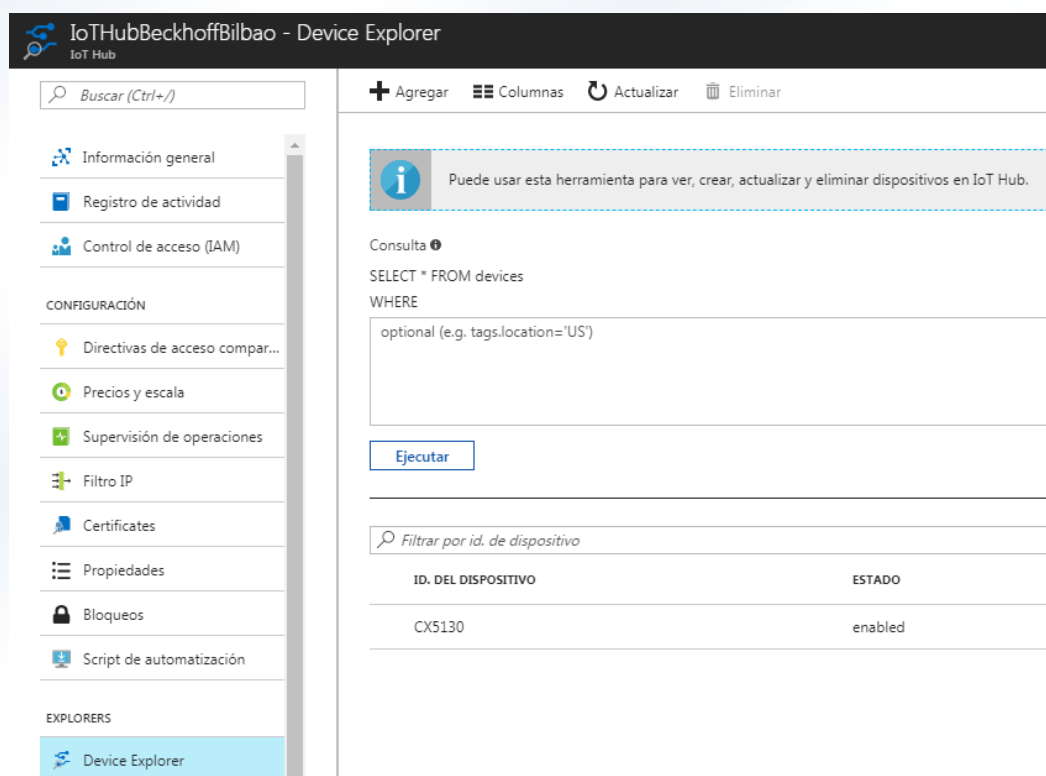
Una vez creado aparecerán las claves de acceso para este dispositivo (más adelante se podrá acceder a estas claves desde la misma aplicación o desde el portal de Azure):



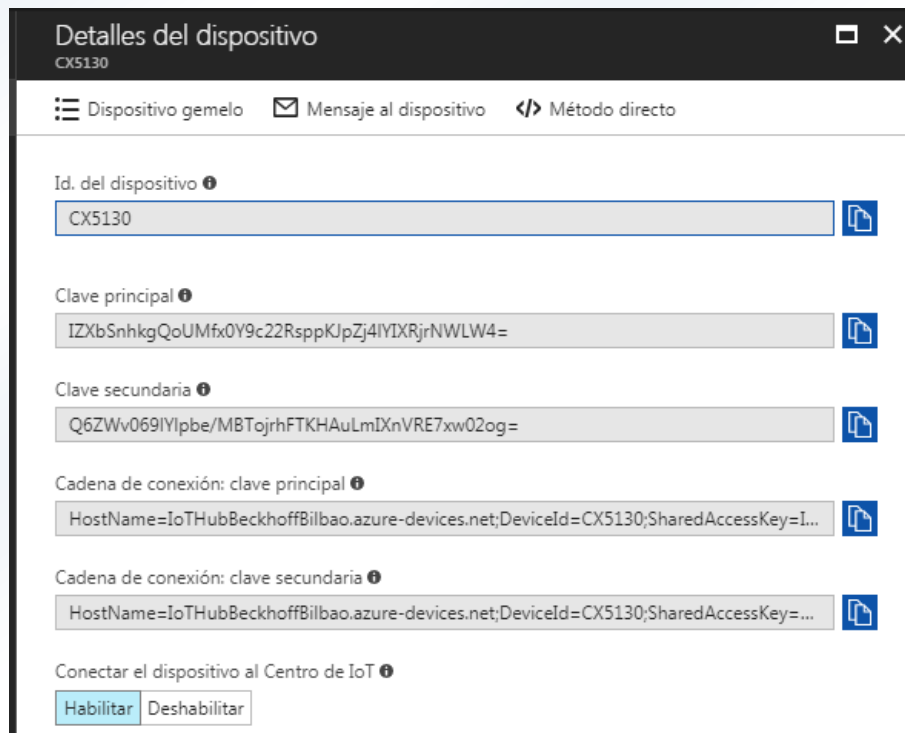
Volviendo a la pestaña de 'Management' y pulsando sobre el botón de 'Refresh' se actualizará la lista de dispositivos creados y debería aparecer el dispositivo recién creado.



Si se accede al centro del IoT Hub desde el portal de Azure y se va a la sección de 'Device Explorer' se podrá ver la información de este dispositivo, habilitarlo o deshabilitarlo, etc...



Si se pincha sobre un dispositivo se pueden ver sus claves de acceso:



Una vez se ha creado el dispositivo ya se puede comunicar con él desde TwinCAT con el suplemento 'TF6701 TC3 IoT Communication' o con el 'TF6720 TC3 IoT Data Agent'.

3. COMUNICAR CON EL TF6701 IOT COMMUNICATION

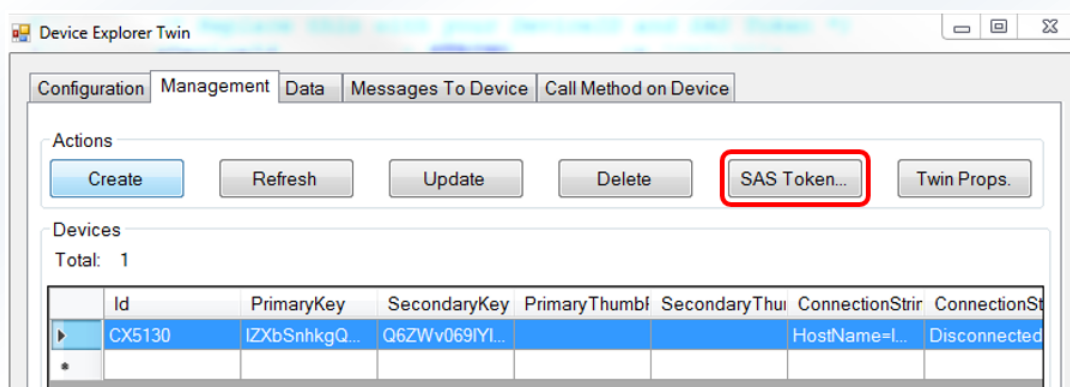
Si se va a comunicar una CPU con TwinCAT con un dispositivo del IoT Hub utilizando el suplemento TF6701 TC3 IoT Communication es necesario hacer unos pasos previos antes de comenzar con el código de PLC, estos pasos son generar un SAS Token para comunicar con el dispositivo y un certificado para la conexión con Azure.

GENERAR EL SAS TOKEN

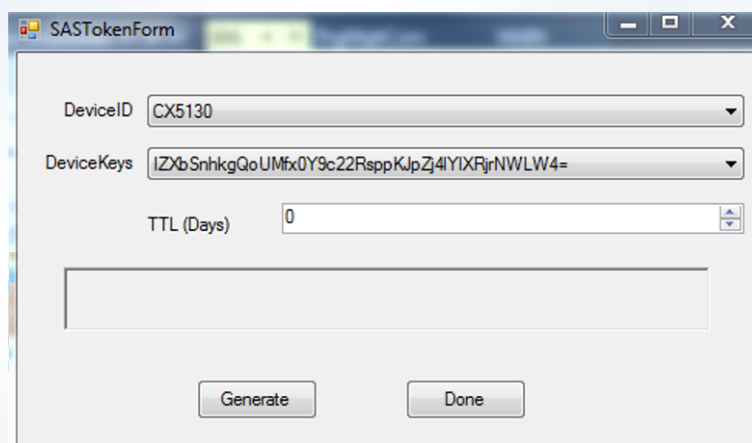
El suplemento TF6701 TC3 IoT Communication comunica por MQTT con los dispositivos de Azure, por lo que se va a utilizar la interfaz MQTT que proporciona cada uno de los dispositivos del centro de IoT Hub ([enlace de información](#)).

Para poder utilizar esta interfaz es necesario generar un SAS Token, que será la cadena de conexión con todos los datos de acceso. Es importante tener en cuenta que cada dispositivo del centro de IoT solo puede tener un único SAS Token (cuando se genera uno nuevo el anterior se elimina) y solo se puede utilizar este SAS Token una vez en cada instante, es decir, no puede haber dos clientes MQTT (en este caso dos CPUs con TwinCAT) utilizando el mismo SAS, este es el motivo por el que cada CPU que use este suplemento deberá atacar a un dispositivo del IoT Hub diferente.

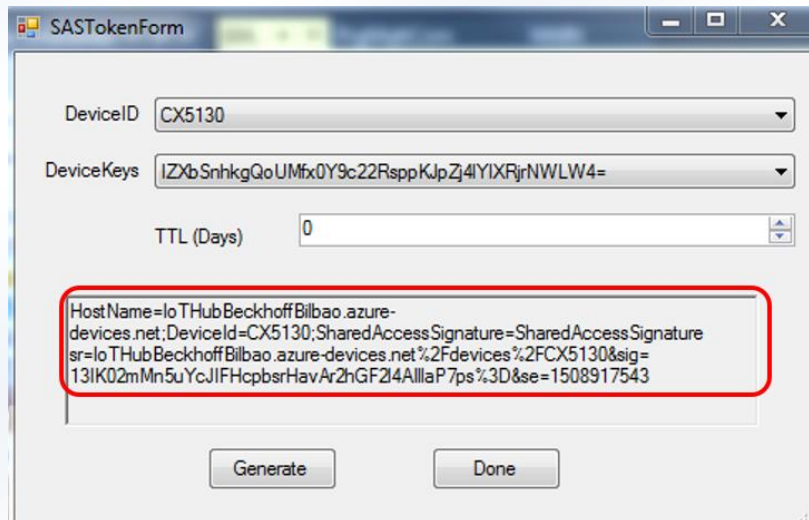
Para generar el SAS Token desde el 'Device Explorer' se debe ir a la pestaña de 'Management', seleccionar el dispositivo y pulsar sobre el botón de 'SAS Token':



En el asistente se puede modificar el número de días durante los que será válido este SAS Token (una vez expire puede haber 5 minutos extra durante los cuales el SAS Token siga siendo válido debido a la gestión en Azure).



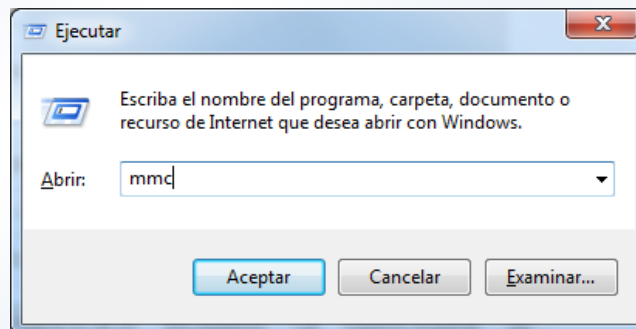
Una vez se pulse sobre el botón de 'Generate' aparecerá el código SAS generado (este código será necesario en el programa de TwinCAT asique hay que copiarlo):



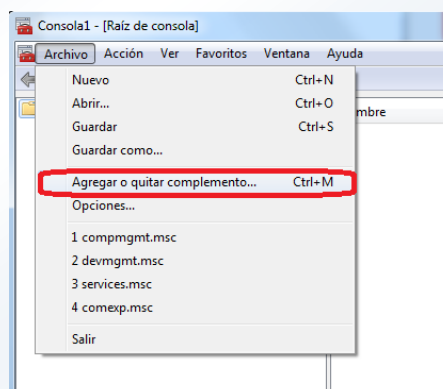
GENERAR EL CERTIFICADO

Para poder comunicar con seguridad con el servicio de Azure es necesario utilizar un certificado de una entidad de certificación que asegure que la conexión se está estableciendo con el servidor de Azure. Para ello se puede utilizar el certificado de Baltimore CyberTrust.

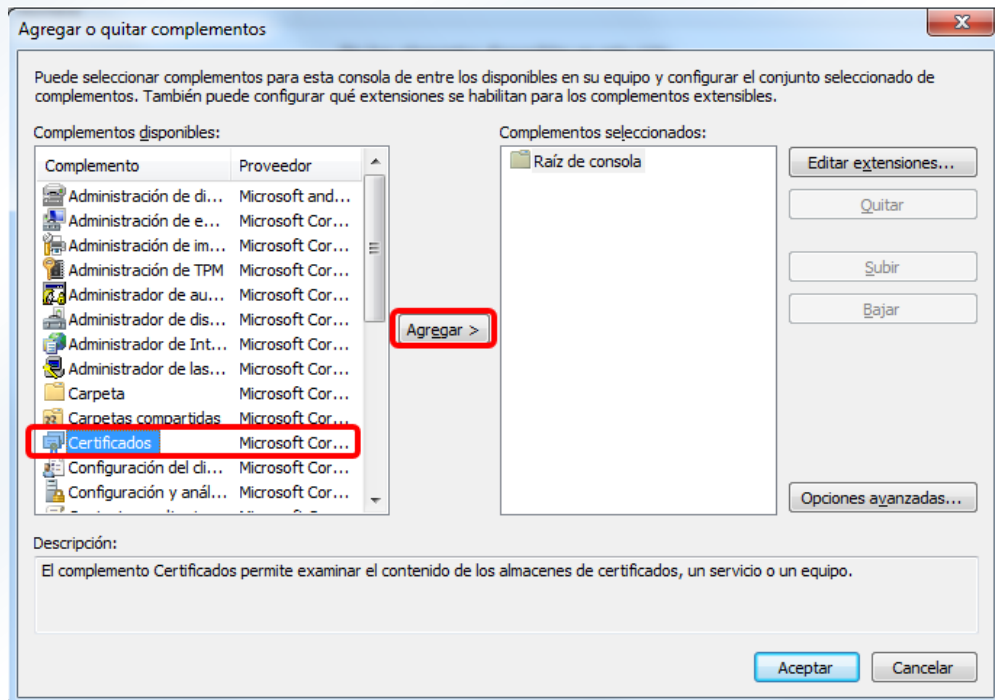
Es posible generar este certificado desde cualquier PC con un sistema operativo Windows utilizando la 'Microsoft Management Console' que se puede abrir con el comando 'mmc' sobre la aplicación de 'Ejecutar' que se puede abrir pulsando 'Windows+R':



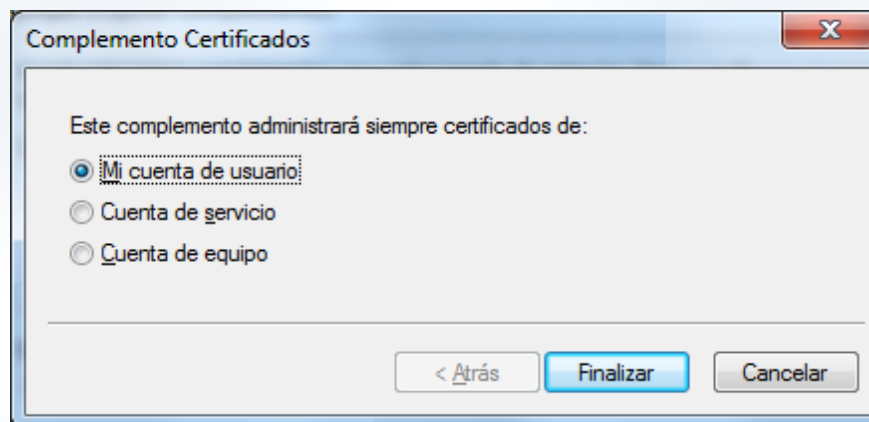
En la aplicación que se abrirá hay que seleccionar la opción de 'Archivo' → 'Agregar o quitar complemento...':



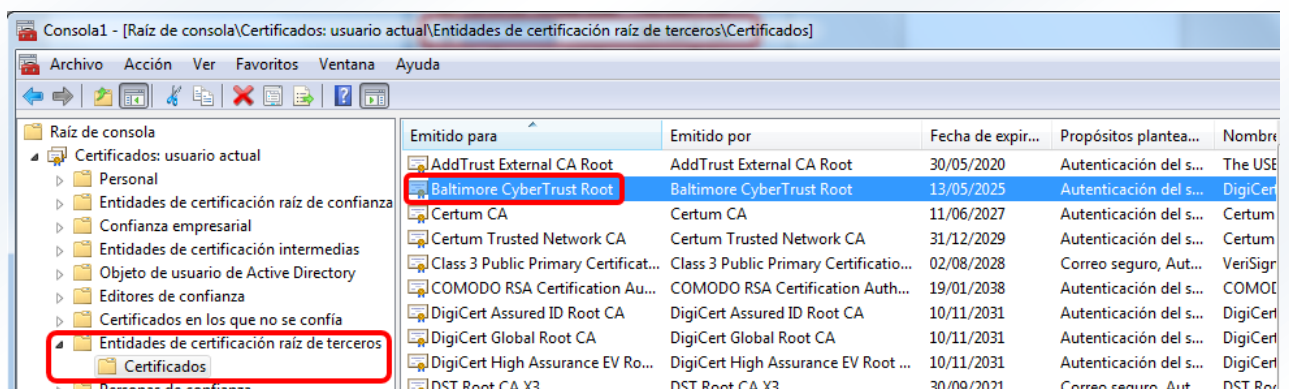
En el asistente que aparece hay que seleccionar el complemento de 'Certificados' y pulsar sobre 'Agregar':



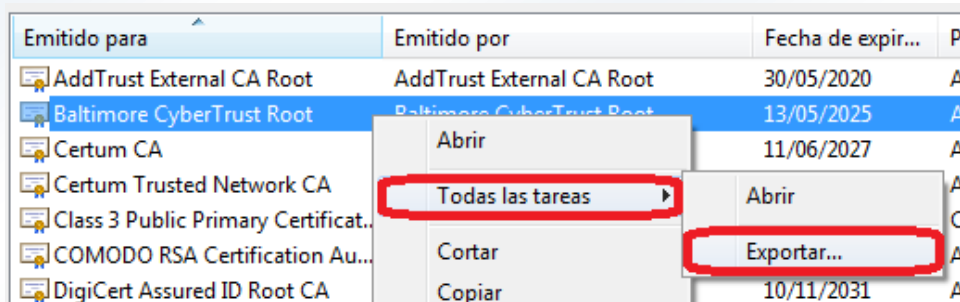
En la siguiente ventana hay que seleccionar la opción de 'Mi cuenta de usuario', ya que como se va a generar el certificado de una entidad de terceros no es relevante la cuenta que se seleccione:



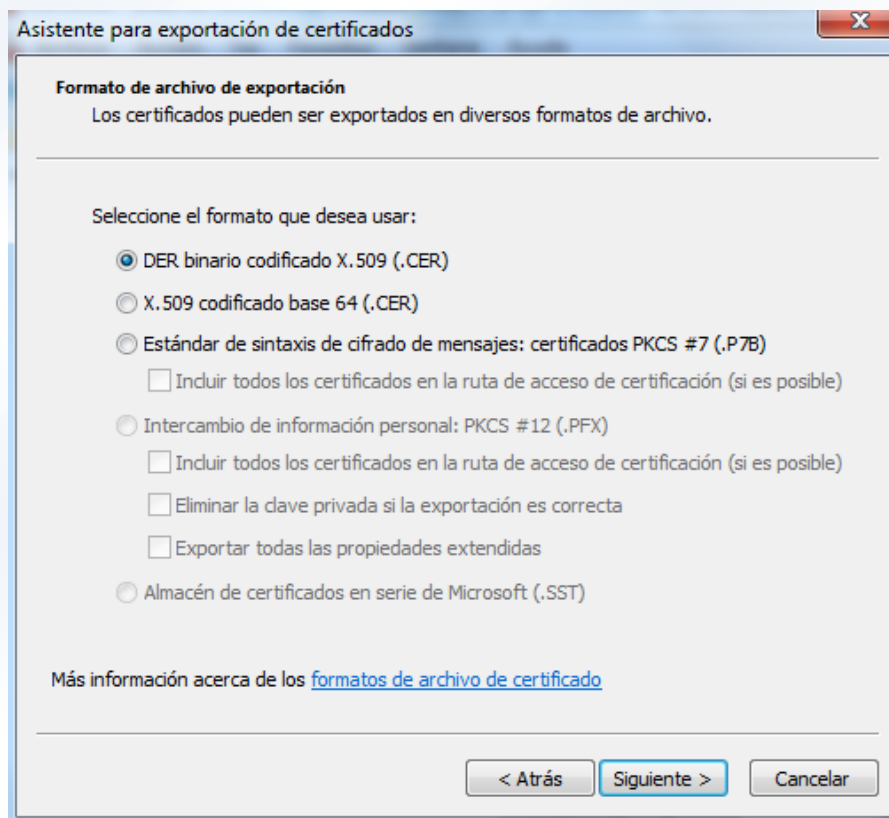
Al darle a 'Finalizar' aparecerán los diferentes certificados disponibles. En este caso el que se va a utilizar es el certificado de 'Baltimore CyberTrust Root':



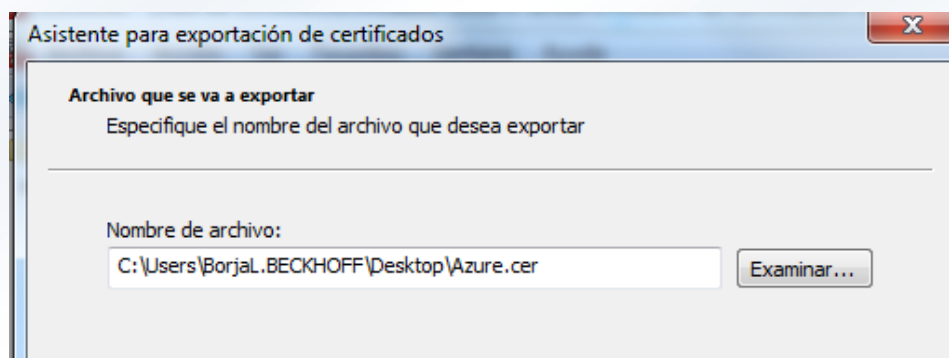
Una vez seleccionado hay que exportarlo pulsando botón derecho sobre el:



En el asistente que aparecerá hay que seguir las instrucciones para exportarlo, Uno de los pasos será seleccionar el formato de salida, se puede elegir 'DER binario codificado X.509 (CER)':



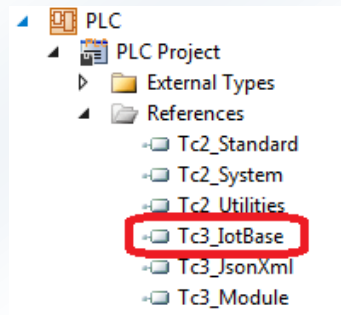
Por ultimo habrá que seleccionar el directorio de salida de este fichero:



Este fichero que se generará es el certificado que habrá que utilizar posteriormente en el programa de TwinCAT.

PROGRAMA DE PLC

Para la comunicación desde TwinCAT PLC con el centro de IoT Hub de Azure es necesario añadir la librería 'Tc3_lotBase' al proyecto:



Conexión con el dispositivo de Azure IoT Hub en TwinCAT



Para poder conectarse al dispositivo de Azure para enviar y recibir mensajes es necesario saber el nombre del dispositivo de Azure, la ruta al certificado de Azure en el equipo y el SAS Token:

```
VAR CONSTANT
    sDeviceId : STRING := 'CX5130';
    sRutaCertificadoAzure : STRING(255) := 'C:\Azure.cer';
    sAzureSasToken : STRING(511) := 'HostName=IoTHubBeckhoffBilbao.azure-
devices.net;DeviceId=CX5130;SharedAccessSignature=SharedAccessSignature
sr=IoTHubBeckhoffBilbao.azure-
devices.net%2Fdevices%2FCX5130&sig=13Lk02mMn5uYcJIFHcpbsrHavAr2hGF2l4A1lllaP7ps%
3D&se=1508917543';
END_VAR
```

Para realizar la comunicación en el programa hay que llamar al FB_lotMqttClient y poner los parámetros del certificado y el SAS Token. El nombre del dispositivo se utiliza para generar el topic al que subscribirse y al que publicar los mensajes. Estos topics tienen una estructura fija que especifica Microsoft Azure y siguen las siguientes reglas:

- Topic al que publicar: 'devices/<NombreDispositivo>/messages/events/readpipe/'
- Topic al que subscribirse: 'devices/<NombreDispositivo>/messages/devicebound/#'

En este caso deberían quedar de la siguiente manera:

| | | |
|---|-------------|--|
|  sTopicAlQuePublicar | STRING(255) | 'devices/CX5130/messages/events/readpipe/' |
|  sTopicSubscrito | STRING(255) | 'devices/CX5130/messages/devicebound/#' |

En el siguiente código se ha generado una función que genera el nombre según esta estructura pasándole como parámetro de entrada el nombre del dispositivo creado en el centro de IoT Hub, pero también se podría haber puesto como una cadena de texto constante.

Por otro lado, en esta inicialización se añade también la instancia al FB que se encargará de desencolar los mensajes que se reciban desde la nube.

Por último, hay que tener en cuenta que la llamada al método 'Execute' debe ser en cada ciclo para mantener la comunicación en segundo plano con el driver de TwinCAT que se encarga de subir los datos, además su parámetro de entrada debe estar a TRUE.


```
IF bInicializacion THEN
    bInicializacion := FALSE;
    // Preparar topics a los que publicar y subscribirse
    sTopicAlQuePublicar := F_CreatePublisherTopic(sDeviceId);
    sTopicSubscrito := F_CreateSubscriberTopic(sDeviceId);
    // Datos para la conexión a Azure IoT Hub por MQTT
    fbMqttClient.stTLS.sCA := sRutaCertificadoAzure;
    fbMqttClient.stTLS.sAzureSas := sAzureSasToken;
    fbMqttClient.ipMessageQueue := fbMessageQueue;
END_IF
fbMqttClient.Execute(bConnectar);
```

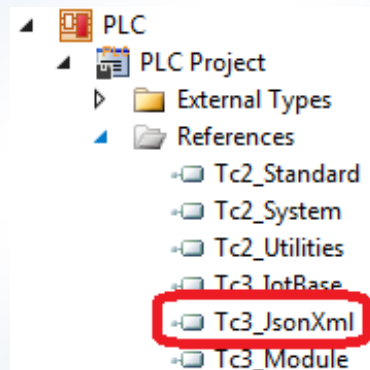
El feedback de si se ha conectado correctamente con el centro de IoT se puede leer de la variable 'bConnected' en el FB_IotMqttClient:

```
IF fbMqttClient.bConnected TRUE THEN
```

Publicar datos del dispositivo a la nube en TwinCAT

La publicación de los datos se puede hacer tanto en formato texto como en datos binarios sin procesamiento, la elección dependerá de la aplicación y del ancho de banda (en texto serán más bytes a enviar).

En este ejemplo se va a enviar un telegrama en formato JSON. Tanto para codificar el telegrama como para decodificarlo está disponible la librería 'Tc3_JsonXml' (aunque al ser una cadena de texto también se podría codificar y decodificar manualmente, aunque sería más pesado).



En esta librería está disponible el 'FB_JsonSaxWriter' que permite ir añadiendo el texto necesario en cada punto al mensaje JSON.

Dentro de este FB hay diferentes métodos que se encargan de modificar el contenido del JSON. El método 'StartObject' sirve para añadir el carácter '{' y el método 'EndObject' para añadir el carácter '}'. Estos métodos se pueden utilizar en cualquier punto para crear mensajes JSON de varios niveles. El método 'AddKey' sirve para añadir el nombre de la variable entre comillas y los dos puntos y el método 'AddReal' para meter el valor de la variable (en este caso una variable de tipo REAL (hay métodos para variables INT, BOOL, etc...)). Por último, el método 'GetDocument' sirve para convertir el objeto JSON a una cadena de texto y guardarla en una variable de tipo STRING.

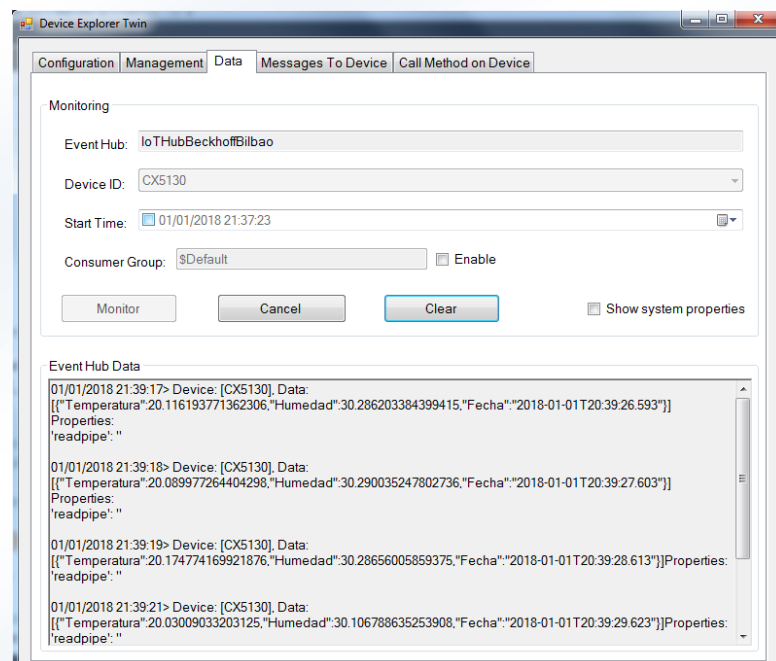
Una vez se tiene el mensaje a enviar formateado hay que llamar al método 'Publish' especificando el topic creado previamente, el mensaje a enviar (a través de la dirección de la variable que contiene el mensaje JSON y su tamaño en bytes) y el nivel de QoS que especificará si se debe esperar un feedback de que se ha recibido el mensaje en la nube al menos una vez, exactamente una vez o no se quiere esperar al feedback (niveles QoS 1, 2 y 0 respectivamente).

```
// Comprobar si se esta conectado
IF fbMqttClient.bConnected THEN
    // Dar la orden de envio de datos cada segundo
    fbTemporizador(IN:=TRUE,PT:=T#1S);
    IF fbTemporizador.Q THEN
        fbTemporizador(IN:=FALSE);
        // Crear telegrama JSON a enviar
        // Vacía el objeto JSON
        fbCrearJson.ResetDocument();
        fbCrearJson();
        // Añade la llave de apertura
        fbCrearJson.StartObject();
        // Añade las parejas clave/valor al mensaje
        fbCrearJson.AddKey('Temperatura');
        fbCrearJson.AddReal(Temperatura);
        fbCrearJson.AddKey('Humedad');
        fbCrearJson.AddReal(Humedad);
        fbCrearJson.AddKey('Fecha');
        fbCrearJson.AddString(Fecha);
        // Añade la llave de clausura
        fbCrearJson.EndObject();
        // Obtiene el STRING con el mensaje JSON
        sJson := fbCrearJson.GetDocument();
        // Publicar el telegrama JSON a la cuenta de Azure
        fbMqttClient.Publish(sTopic := sTopicAlQuePublicar,
            pPayload:= ADR(sJson), nPayloadSize:= TO_UDINT(LEN(sJson)),
            eQoS:= TcIoTmqttQos.AtMostOnceDelivery, bRetain:= FALSE,
            bQueue:= FALSE );
    END_IF
END_IF
```

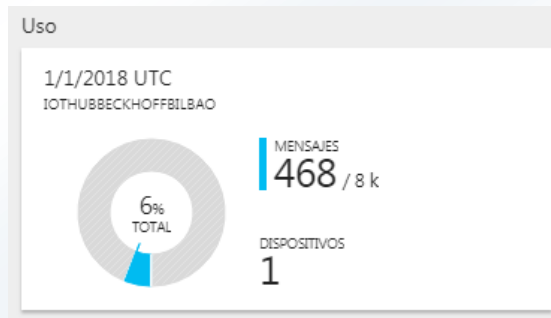
Al ejecutarse este código se habrá generado un mensaje JSON con el siguiente formato:

| | | |
|-------|-------------|---|
| sJson | STRING(255) | '{"Temperatura":20.094158172607423,"Humedad":30.275848388671876,"Fecha":"2018-01-01T20:39:30.633"}' |
|-------|-------------|---|

Si se quieren monitorizar los mensajes que están llegando al centro de IoT se puede hacer a través de la aplicación 'Device Explorer' en la pestaña 'Data'. En esta pestaña se puede seleccionar el dispositivo que se quiere monitorizar y al darle al botón de 'Monitor' se empezarán a recibir los mensajes:



Por otro lado, en el panel de control de IoT Hub en el portal de Azure se pueden ver los mensajes consumidos.



Lectura de mensajes de nube a dispositivo en TwinCAT

Para la recepción de los datos, en primer lugar, es necesario subscribirse al topic para poder recibir los mensajes que lleguen de la nube al dispositivo.

```
// Comprobar si se esta conectado
IF fbMqttClient.bConnected THEN
    // Subscribirse al topic
    IF NOT bSubscrito THEN
        bSubscrito := fbMqttClient.Subscribe(sTopic:=sTopicSubscrito,
        eQoS:=TcIotMqttQos.AtMostOnceDelivery);
    ELSE
        // En el arranque se puede enviar un mensaje de inicialización, por
        ejemplo la localización de la CPU o el estado
    END_IF
END_IF
```

Una vez se está suscrito al topic hay que monitorizar si hay algún mensaje recibido en la cola y, de ser así, desencolarlo. Una vez se ha desencolado se puede obtener el topic desde el que se ha recibido el mensaje, así como el propio mensaje.

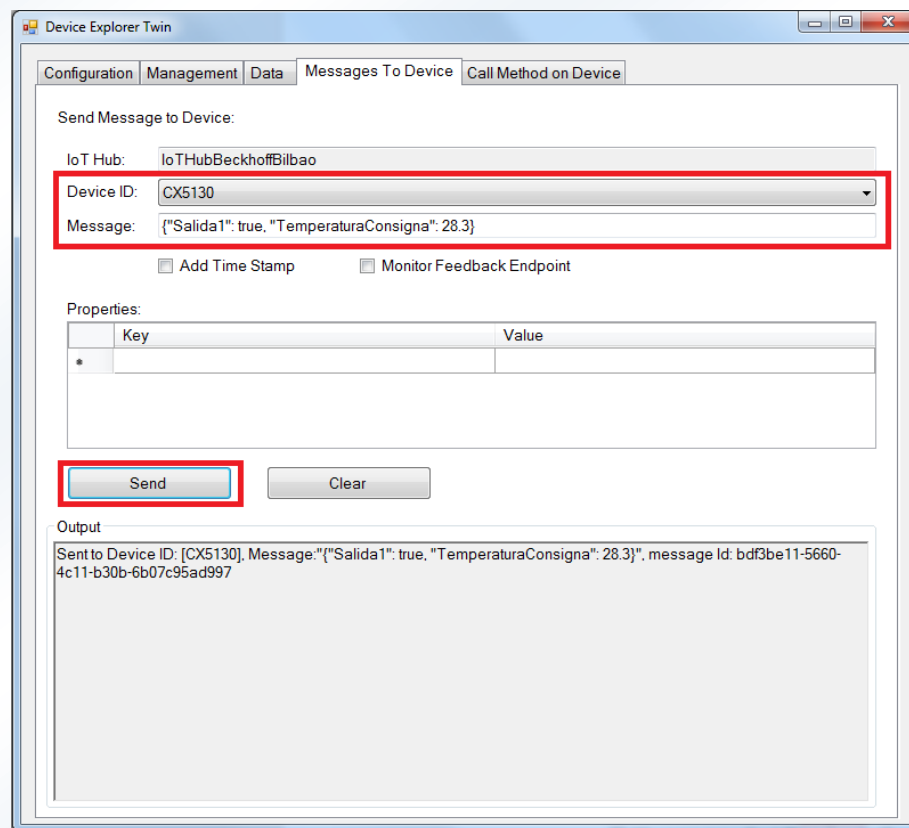
En caso de que el mensaje tenga el formato JSON se puede utilizar el 'FB_JsonDomParser' para comprobar las claves que hay en el mensaje, el tipo de dato recibido y, en caso de que se cumplan las condiciones necesarias, leer ese valor.

En el siguiente ejemplo se leen los valores de un mensaje recibido que tendría el formato '{"Salida1": true, "TemperaturaConsigna": 28.3}'. Tener en cuenta que para que se detecten las variables de tipo BOOL correctamente el valor debe estar escrito en minúsculas y sin comillas de ningún tipo (es decir o 'true' o false' sin las comillas).

```
// Esperar a que haya algun mensaje en la cola
IF fbMessageQueue.nQueuedMessages > 0 THEN
    // Desencolar el mensaje
    IF fbMessageQueue.Dequeue(fbMessage:=fbMessage) THEN
        // Obtiene el topic en el que se ha recibido el mensaje
        fbMessage.GetTopic(pTopic:=ADR(sTopicDatosRecibidos),
        nTopicSize:=SIZEOF(sTopicDatosRecibidos) );
        // Obtiene el mensaje
        fbMessage.GetPayload(pPayload:=ADR(sMensajeRecibido),
        nPayloadSize:=SIZEOF(sMensajeRecibido), bSetNullTermination:=FALSE);
        // Decodificar el mensaje JSON
        // En primer lugar se convierte la cadena de texto a un objeto JSON
        jsonDoc := fbDecodificarJson.ParseDocument(sMensajeRecibido);
        // Busca la clave dentro del JSON
        jsonProp := fbDecodificarJson.FindMember(jsonDoc, 'TemperaturaConsigna');
```

```
// Si hay una clave con el nombre 'TemperaturaConsigna' se lee su valor
IF (jsonProp <> 0) THEN
    // Comprueba el tipo de dato y si coincide se lee su valor
    IF fbDecodificarJson.IsDouble(jsonProp) THEN
        nOrdenAnalogica:=fbDecodificarJson.GetDouble(jsonProp);
    END_IF
END_IF
jsonProp := fbDecodificarJson.FindMember(jsonDoc, 'Salida1');
IF (jsonProp <> 0) THEN
    IF fbDecodificarJson.IsBool(jsonProp) THEN
        bOrdenesSalida1 := fbDecodificarJson.GetBool(jsonProp);
    END_IF
END_IF
END_IF
END_IF
```

Para probar este código se puede mandar un mensaje desde la nube al dispositivo con ayuda de la aplicación 'Device Explorer'. En la pestaña 'Message to Device' se puede seleccionar el dispositivo al que se va a mandar el mensaje y escribir el texto que se va a enviar.



Una vez enviado, en TwinCAT se puede ver que se recibe correctamente el mensaje y se decodifica el valor de sus variables correctamente:

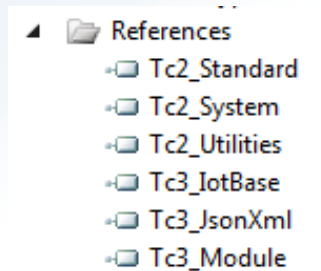
| | | |
|------------------|-------------|--|
| sMensajeRecibido | STRING(255) | '{"Salida1": true, "TemperaturaConsigna": 28.3}' |
| bOrdenesSalida1 | BOOL | TRUE |
| nOrdenAnalogica | LREAL | 28.3 |

En caso de que se estuviesen enviando datos binarios el código sería el mismo, lo único que habría que cambiar es la variable donde se guarda el 'payload' recibido que, en vez de ser una variable de tipo STRING, sería una estructura que contendría los campos necesarios para interpretar correctamente los datos recibidos.

Código de TwinCAT completo

En este apartado se puede ver el código completo de la comunicación desde TwinCAT con un dispositivo del centro de IoT Hub de forma que se pueda copiar de una forma más rápida y sencilla.

En primer lugar, antes de copiar el código, el proyecto de PLC debe tener añadidas las siguientes librerías:



La declaración de variables es la siguiente (recordar modificar las variables constantes con la información del dispositivo con el que se vaya a comunicar).

```
PROGRAM PRG_AzurePorMQTT
VAR
    bInicializacion : BOOL := TRUE;
    bConnectar : BOOL := TRUE;
    // Cliente MQTT
    fbMqttClient : FB_IotMqttClient;
    // Mensaje JSON
    sJson : STRING(255);
    fbCrearJson : FB_JsonSaxWriter;
    // Variables para publicar mensajes de dispositivo a nube
    sTopicAlQuePublicar : STRING(255);
    nMensajesEnviados : UDINT;
    fbTemporizador : TON;
    // Variables para recibir mensajes de nube a dispositivo
    bSubscrito : BOOL;
    sTopicSubscrito : STRING(255);
    {attribute 'TcEncoding':='UTF-8'}
    sTopicDatosRecibidos : STRING(255);
    {attribute 'TcEncoding':='UTF-8'}
    sMensajeRecibido : STRING(255);
    fbMessageQueue : FB_IotMqttMessageQueue;
    fbMessage : FB_IotMqttMessage;
    // Decodificar JSON
    jsonDoc : SJsonValue;
    fbDecodificarJson : FB_JsonDomParser;
    jsonProp : SJsonValue;
    // Variables recibidas desde la nube
    bOrdenesSalida1 : BOOL;
    nOrdenAnalogica : LREAL;
    // Variables para la simulación de datos a subir a la nube
    Temperatura : LREAL;
    Humedad : LREAL;
    Fecha: STRING;
    // Obtener fecha del sistema
    Ticks : T_FILETIME;
    fbObtenerTicks : GETCPUOUNTER;
    // Generar valores aleatorios
    fbRandom : DRAND;
END_VAR
```

```
VAR CONSTANT
    sDeviceId : STRING := 'CX5130';
    sRutaCertificadoAzure : STRING(255) := 'C:\Azure.cer';
    sAzureSasToken : STRING(511) := 'HostName=IoTHubBeckhoffBilbao.azure-
devices.net;DeviceId=CX5130;SharedAccessSignature=SharedAccessSignature
sr=IoTHubBeckhoffBilbao.azure-
devices.net%2Fdevices%2FCX5130&sig=13Lk02mMn5uYcJIFHcpbsrHavAr2hGF2l4Al1laP7ps%
3D&se=1508917543';
END_VAR
```

El código, en texto estructurado, es el siguiente:

```
IF bInicializacion THEN
    bInicializacion := FALSE;
    // Preparar topics a los que publicar y subscribirse
    sTopicAlQuePublicar := 'devices/CX5130/messages/events/readpipe/';
    sTopicSubscrito := 'devices/CX5130/messages/devicebound/#';
    // Datos para la conexión a Azure IoT Hub por MQTT
    fbMqttClient.stTLS.sCA := sRutaCertificadoAzure;
    fbMqttClient.stTLS.sAzureSas := sAzureSasToken;
    fbMqttClient.ipMessageQueue := fbMessageQueue;
END_IF
fbMqttClient.Execute(bConnectar);

// Simulación de variables para subir a la nube
// Leer hora del sistema
fbObtenerTicks(cpuCntLoDW=> Ticks.dwLowDateTime, cpuCntHiDW=>
Ticks.dwHighDateTime);
Fecha := SYSTEMTIME_TO_STRING(FILETIME_TO_SYSTEMTIME(Ticks));
Fecha[10] := F_ToASC('T');
// Cálculo de señales analógicas a partir de valores aleatorios
fbRandom(Seed := 1);
Temperatura := (fbRandom.Num) * 20.0 + 20;
fbRandom(Seed := 2);
Humedad := (fbRandom.Num) * 30 + 30;

// Comprobar si se está conectado
IF fbMqttClient.bConnected THEN
    // Suscribirse al topic
    IF NOT bSubscrito THEN
        bSubscrito := fbMqttClient.Subscribe(sTopic:=sTopicSubscrito,
eQoS:=TcIotMqttQos.AtMostOnceDelivery);
    ELSE
        // En el arranque se puede enviar un mensaje de inicialización, por
ejemplo la localización de la CPU o el estado
    END_IF
    // Dar la orden de envío de datos cada segundo
    fbTemporizador(IN:=TRUE, PT:=T#1S);
    IF fbTemporizador.Q THEN
        fbTemporizador(IN:=FALSE);
        // Crear telegrama JSON a enviar
        // Vacía el objeto JSON
        fbCrearJson.ResetDocument();
        fbCrearJson();
        // Añade la llave de apertura
        fbCrearJson.StartObject();
        // Añade las parejas clave/valor al mensaje
        fbCrearJson.AddKey('Temperatura');
        fbCrearJson.AddReal(Temperatura);
        fbCrearJson.AddKey('Humedad');
        fbCrearJson.AddReal(Humedad);
        fbCrearJson.AddKey('Fecha');
```

```
fbCrearJson.AddString(Fecha);
// Añade la llave de clausura
fbCrearJson.EndObject();
// Obtiene el STRING con el mensaje JSON
sJson := fbCrearJson.GetDocument();
// Publicar el telegrama JSON a la cuenta de Azure
fbMqttClient.Publish(sTopic := sTopicAlQuePublicar,
pPayload:= ADR(sJson), nPayloadSize:= TO_UDINT(LEN(sJson)),
eQoS:= TcIotMqttQos.AtMostOnceDelivery, bRetain:= FALSE,
bQueue:= FALSE );
// Llevar la cuenta de los mensajes publicados
nMensajesEnviados := nMensajesEnviados + 1;
END_IF
END_IF

// Esperar a que haya algun mensaje en la cola
IF fbMessageQueue.nQueuedMessages > 0 THEN
// Desencolar el mensaje
IF fbMessageQueue.Dequeue(fbMessage:=fbMessage) THEN
// Obtiene el topic en el que se ha recibido el mensaje
fbMessage.GetTopic(pTopic:=ADR(sTopicDatosRecibidos),
nTopicSize:=SIZEOF(sTopicDatosRecibidos) );
// Obtiene el mensaje
fbMessage.GetPayload(pPayload:=ADR(sMensajeRecibido),
nPayloadSize:=SIZEOF(sMensajeRecibido), bSetNullTermination:=FALSE);
// Decodificar el mensaje JSON
// En primer lugar se convierte la cadena de texto a un objeto JSON
jsonDoc := fbDecodificarJson.ParseDocument(sMensajeRecibido);
// Busca la clave dentro del JSON
jsonProp := fbDecodificarJson.FindMember(jsonDoc, 'TemperaturaConsigna');
// Si hay una clave con el nombre 'TemperaturaConsigna' se lee su valor
IF (jsonProp <> 0) THEN
// Comprueba el tipo de dato y si coincide se lee su valor
IF fbDecodificarJson.IsDouble(jsonProp) THEN
nOrdenAnalogica:=fbDecodificarJson.GetDouble(jsonProp);
END_IF
END_IF
jsonProp := fbDecodificarJson.FindMember(jsonDoc, 'Salida1');
IF (jsonProp <> 0) THEN
IF fbDecodificarJson.IsBool(jsonProp) THEN
bOrdenesSalida1 := fbDecodificarJson.GetBool(jsonProp);
END_IF
END_IF
END_IF
END_IF
```

4. COMUNICAR CON EL TF6720 TC3 IOT DATA AGENT

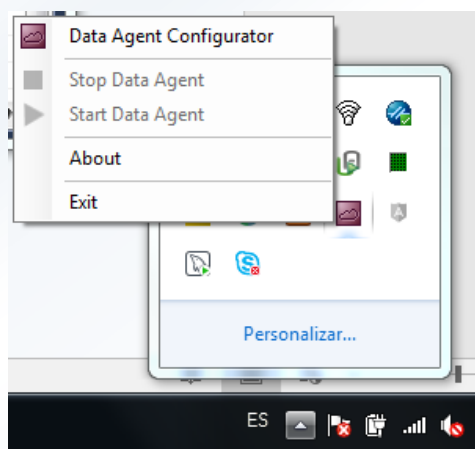
El suplemento TF6720 TC3 IoT Data Agent permite subir los datos a la nube sin necesidad de programar nada en el programa de PLC de TwinCAT. Es un servicio que corre sobre Windows (no se ejecuta en el entorno de tiempo real) que permite recoger datos por ADS (para comunicar con equipos de Beckhoff con TwinCAT 2 o TwinCAT 3) o por OPC UA (para equipos de terceros) y los puede subir a Microsoft Azure, Amazon Web Services o a un broker MQTT genérico.



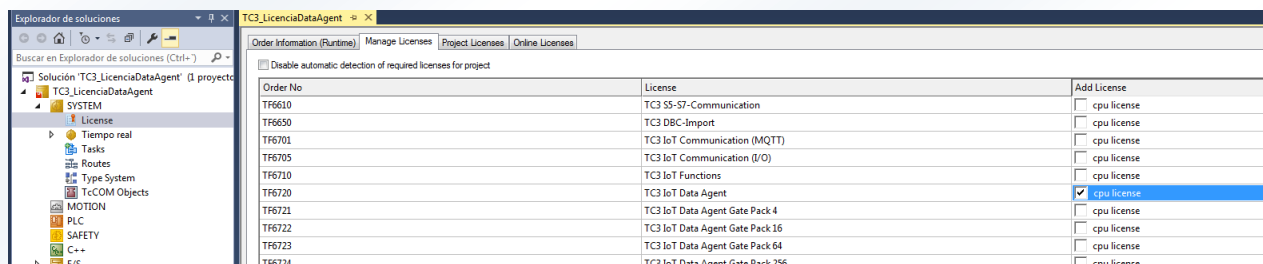
Este suplemento no se instala automáticamente con TwinCAT por lo que hay que ir a la [página web de Beckhoff](#) a descargarse el ejecutable para la instalación.

| | | |
|--|-----------------|--|
|  TF6720 TC3 IoT Data Agent | 1.1.17.0 | The TC3 IoT Data Agent provides IoT communication functions bi-directional in the form of a gateway application that can be configured and operated independently from the TwinCAT real-time environment. Documentation |
|--|-----------------|--|

Una vez se ha instalado el suplemento (con permisos de Administrador) aparecerá en la barra de Windows el icono del Data Agent con acciones para abrir el configurador o para iniciar y parar el servicio.



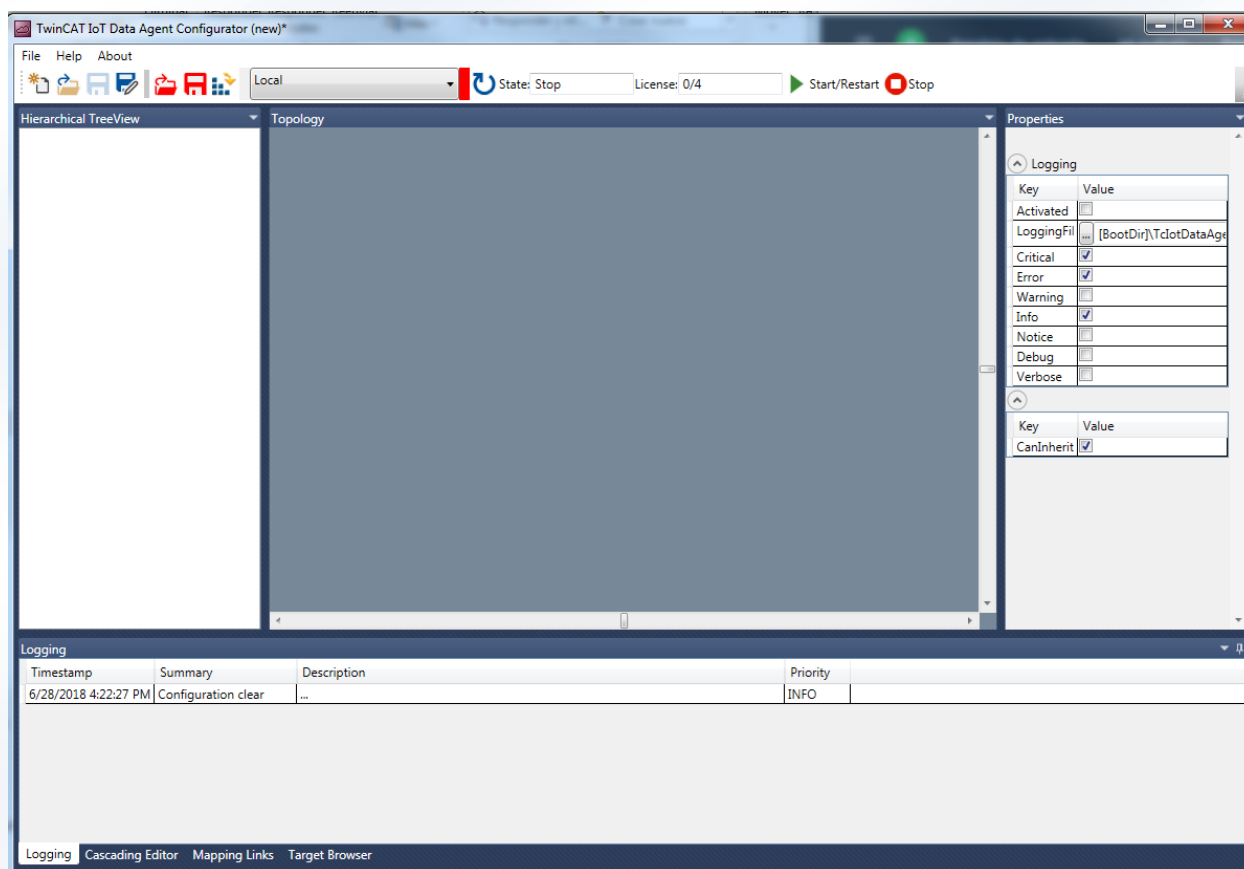
Para este suplemento no es necesaria ninguna licencia de TwinCAT IO ni TwinCAT PLC, únicamente se necesita la licencia propia del Data Agent. Para activar esta licencia hay que generar un proyecto de TwinCAT y añadir manualmente la licencia para poder activarla (bien con la licencia trial de 7 días bien con la definitiva).



Como se puede ver, en las licencias hay diferentes paquetes en función del número de equipos con los que se vaya a comunicar (número de 'Gates'). El Data Agent puede comunicar con equipos por ADS, OPC UA, MQTT, Azure IoT Hub y Amazon Web Service, pero para las licencias únicamente cuentan los equipos de ADS y OPC UA.

Por ejemplo, si se quisiesen leer datos de 3 equipos por OPC UA y de 1 por ADS y se quisiesen subir los datos a una cuenta de Azure, a una de Amazon y a 3 brokers MQTT distintos sería suficiente con la licencia base del Data Agent TF6720 ya que incluye 4 conexiones. Si se incluyesen más equipos, bien de tipo OPC UA bien de ADS, sería necesario añadir las licencias TF6721 – TF6724.

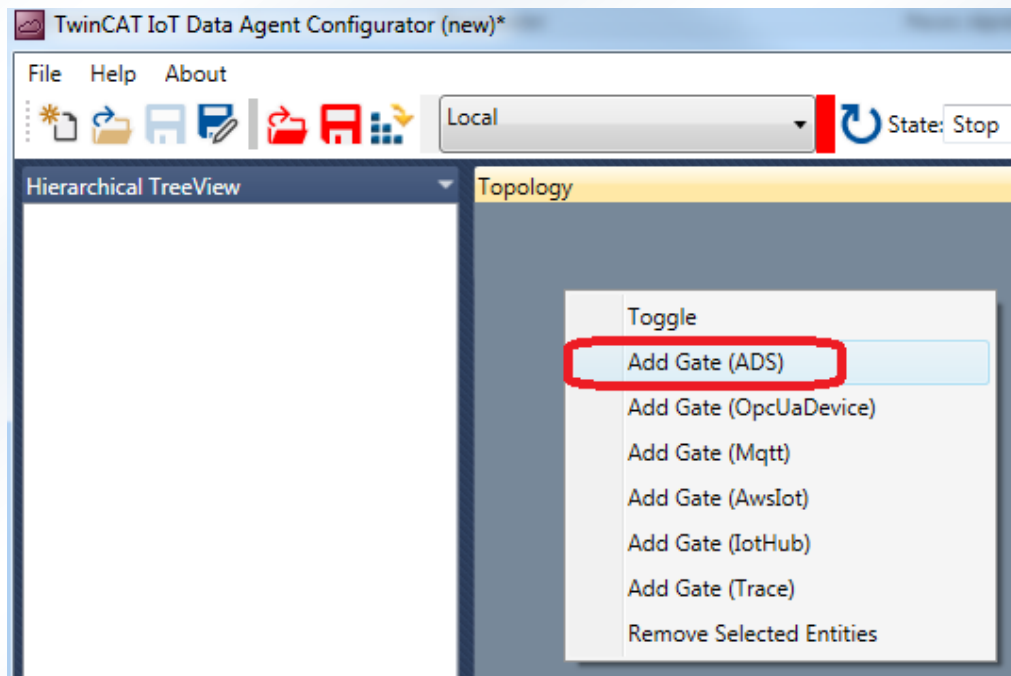
Para crear la configuración, lo primero que hay que hacer es abrir el 'Data Agent Configurator', que es una aplicación que permite configurar las variables que se van a recoger y a que nube se van a enviar. Toda esta configuración se traducirá a un fichero '.xml' de configuración que es el que realmente utiliza el servicio (aunque es transparente para el usuario). Al abrir el configurador se verá la siguiente pantalla.



En esta pantalla se puede ver a qué equipo se está conectado (el equipo en el que se descargará la configuración), el estado del Data Agent, el número de licencias en uso y disponibles, y acciones para iniciar y detener el servicio.

Además, se puede ver la configuración definida en el árbol de la izquierda (en la imagen se puede ver una configuración vacía). En la parte central (la de 'Topology') será donde se haga la configuración de las relaciones entre los diferentes 'Gates'.

En este primer ejemplo se van a leer datos de un equipo con TwinCAT y se van a subir a un dispositivo de Azure IoT Hub, por lo que, como 'Gate' de entrada se utilizará el protocolo ADS.

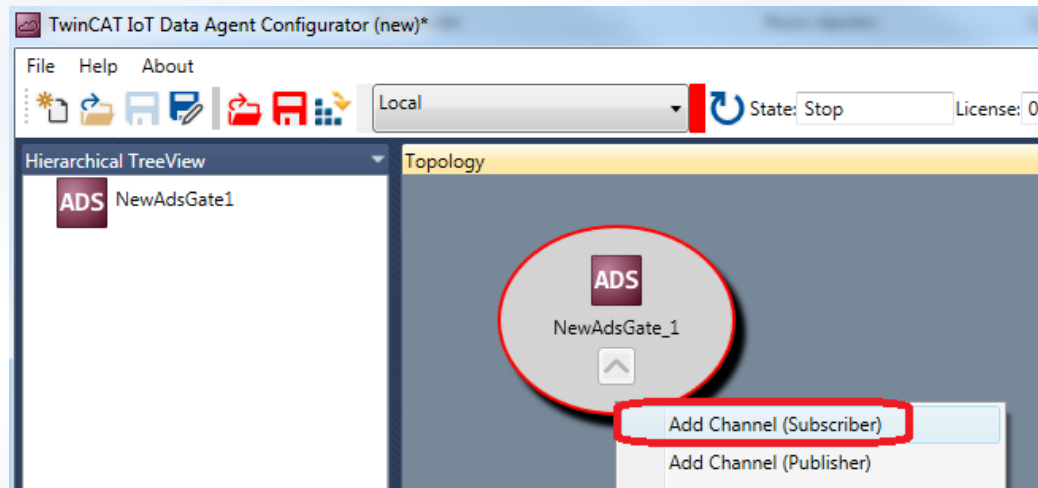


Al añadir el 'Gate' de ADS aparecerá una ventana donde seleccionar el AMS Net Id del equipo al que se va a conectar de entre las rutas que haya configuradas y del puerto del que se van a leer las variables. En este caso, el programa está corriendo en el equipo local y al ser TwinCAT 3 el puerto del programa de PLC será el '851'.



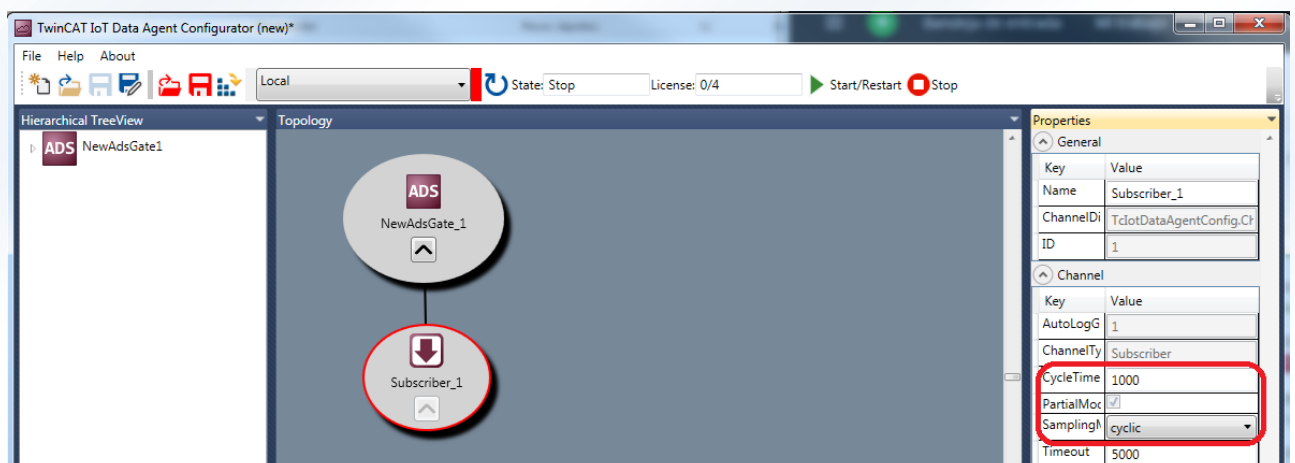
La opción de 'IoMode' se refiere al modo de leer los datos. En modo 'Batched' intenta leer los símbolos agrupando comandos (lo que optimiza la comunicación ADS). En modo 'Direct' lee cada símbolo con un comando independiente (menos optimo, pero obligatorio para la lectura de datos desde BCs).

Una vez creado el 'Gate' (se verá que ya aparece en el árbol de la izquierda), hay que añadirle el canal 'Subscriber' que se encargará de leer las variables en este 'Gate'. Para ello hay que pulsar botón derecho sobre él y seleccionar 'Add Channel (Subscriber)'.

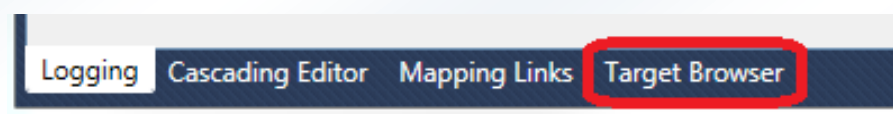


Al añadirlo se pueden ver las opciones de modo de recogida de datos del 'Subscriber', que puede ser:

- **'Cyclic'**: se leerán/escribirán los datos cíclicamente con el ciclo definido en el campo 'CycleTime' (en milisegundos).
- **'OnChange'**: con esta opción se leen únicamente las variables en el instante en que cambien de valor.
- **'ByTrigger'**: (solo en los bloques 'Publisher') se puede definir una condición para que se suban los datos. Esta condición estará vinculada a una variable de lectura del 'Gate'. Este caso se puede usar, por ejemplo, para subir los datos cuando una variable de medida sea mayor a un valor predefinido. La configuración de este trigger se define con los campos de la sección 'TriggerSymbol'.



El último paso en la configuración de este 'Gate' son las variables que se van a leer. Para ello hay que ir a la barra de secciones de abajo y seleccionar el 'Target Browser'.



En el 'Target Browser' aparecen los equipos de los que se haya añadido la ruta de ADS y se puede navegar para ver las variables que tienen sus programas.

Target Browser

ADS OpcUa

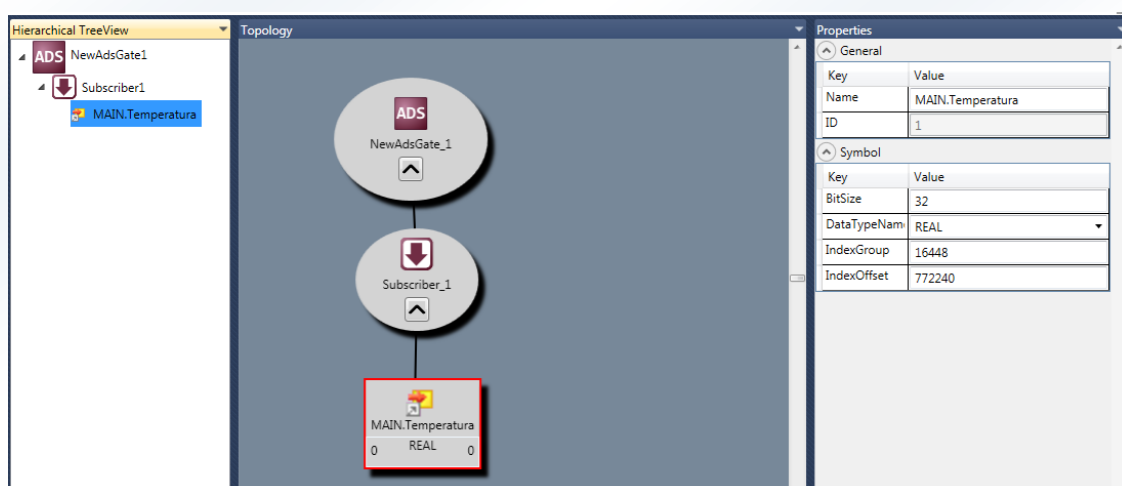
Enter Filter...

| Name | Type | Size | Category | Full-Name | Comment | Subitems | Unit | Context-Mask | Index-Group | Index-Offset | Attributes (Inst) |
|----------------------|------|------|-----------|-------------|---------|----------|------|--------------|-------------|--------------|-------------------|
| Constants | | 0 | Struct | Constants | | 9 | | 0 | 0 | 0 | none |
| MAIN | | 0 | Struct | MAIN | | 1 | | 0 | 0 | 0 | none |
| Temperatura | REAL | 4 | Primitive | MAIN.Te... | | 0 | | 0 | 4040 | BC890 | none |
| TwinCAT_SystemInfoVa | | 0 | Struct | TwinCAT_... | | 5 | | 0 | 0 | 0 | none |

Ports

- BORJAL-NB01
- 350: PlcTask
- 851: Port851
- CX-0A75E0
- CP-2FC9F2
- CX-39A76E
- CX-0A75BC

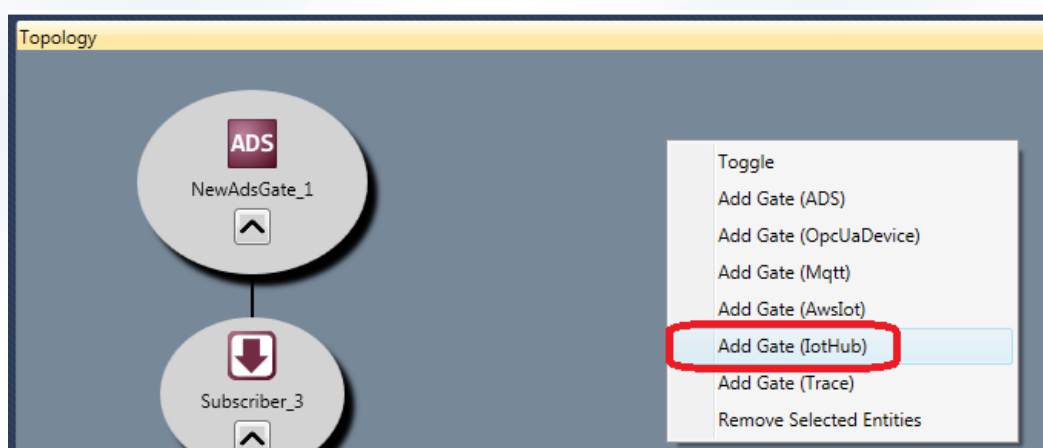
Cuando estén localizadas las variables que se quieren recoger del equipo hay que arrastrarlas al bloque 'Subscriber' con un movimiento de tipo drag&drop.



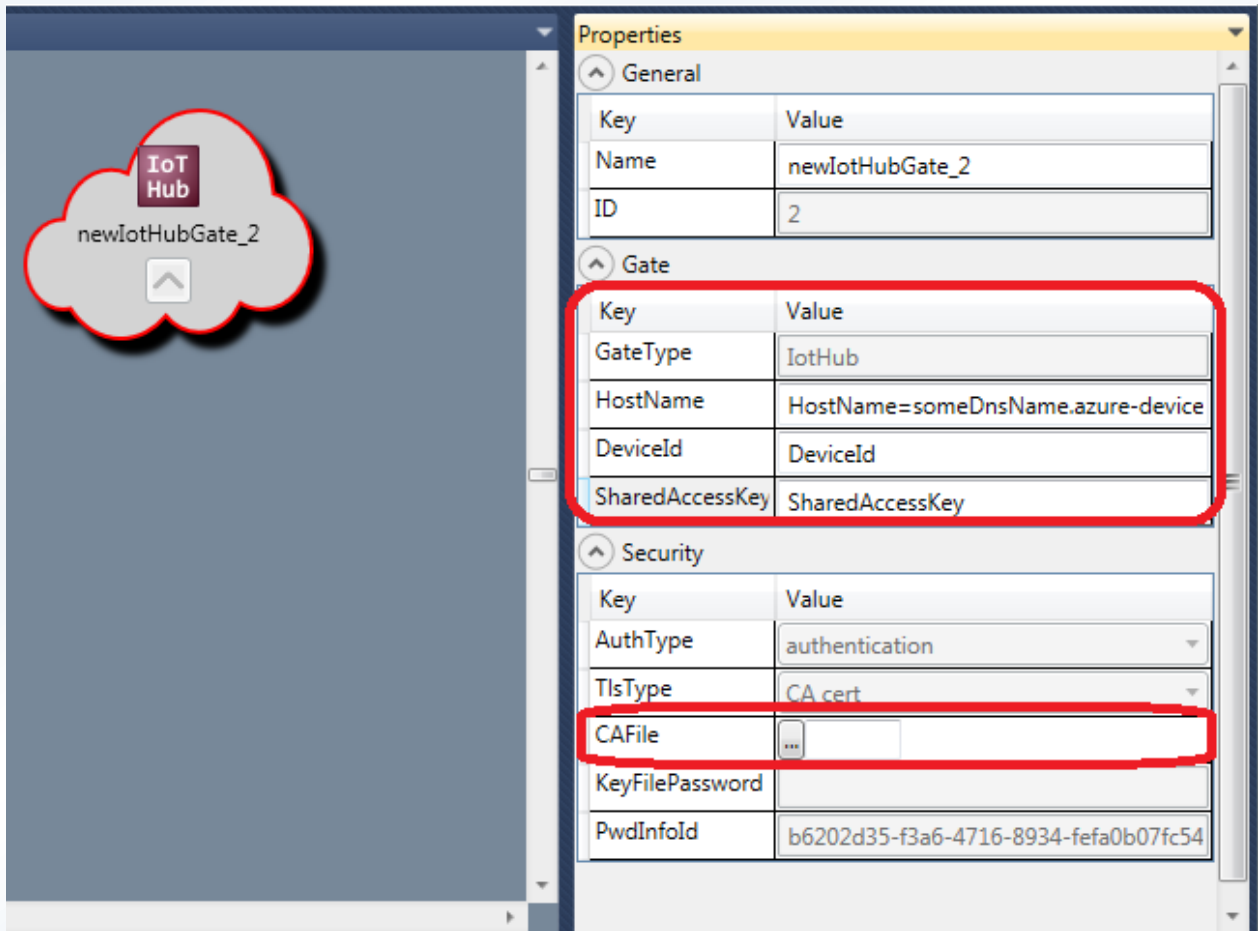
Al arrastrarla con un drag&drop se verá cómo se dibuja un bloque para cada variable. En sus propiedades está toda la información del tipo de dato y de acceso de lectura ADS (estos datos también se pueden añadir manualmente en caso de que no se tenga acceso al equipo del que se quieren recoger los datos).

Al añadir esta variable ya se habría terminado de hacer una configuración mínima de este 'Gate'. Se podrían añadir tantas variables como se necesitasen al mismo 'Subscriber' e, incluso, se podrían añadir diferentes 'Subscriber' al mismo 'Gate', por ejemplo, para subir unas variables a una frecuencia fija, otras por cambio, etc...

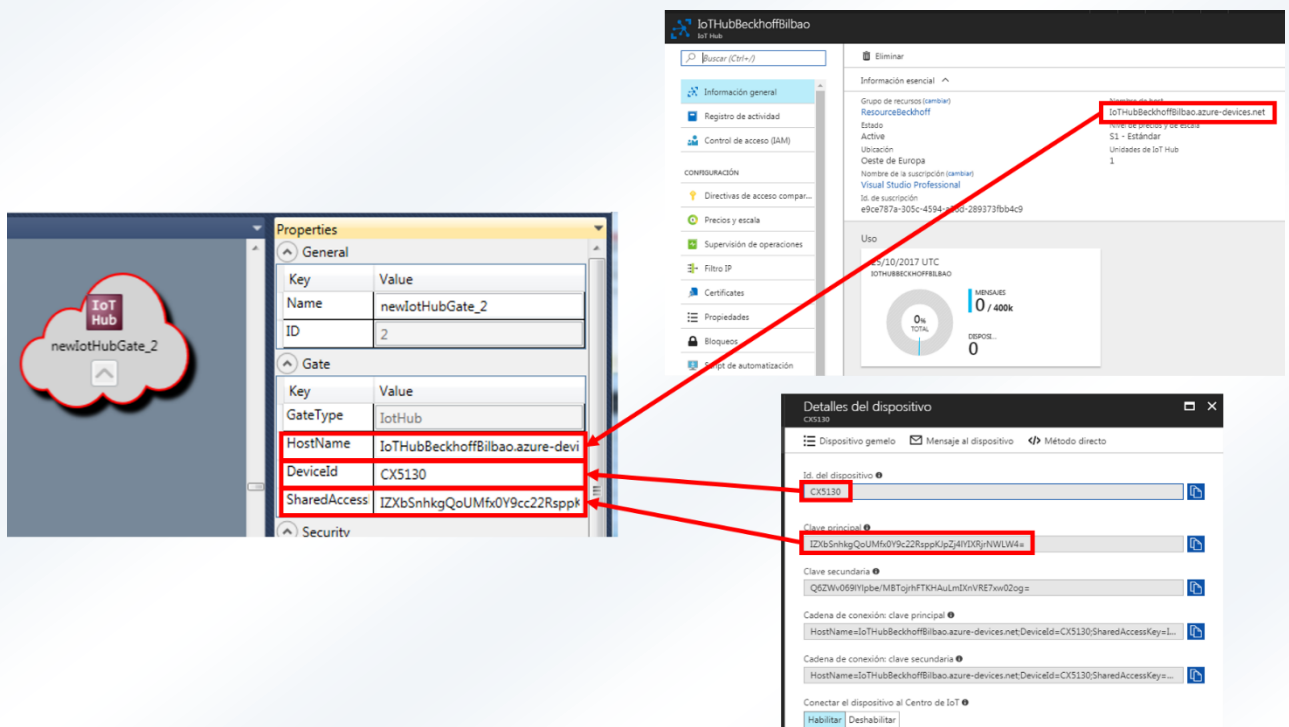
En este punto ya estaría configurado el 'Gate' de ADS, lo que quedaría es añadir el 'Gate' para la comunicación con el dispositivo de Azure y vincular las variables. Para añadir este 'Gate' se siguen los mismos pasos que para añadir el 'Gate' de ADS, pero seleccionando el tipo de 'Azure IoT Hub':



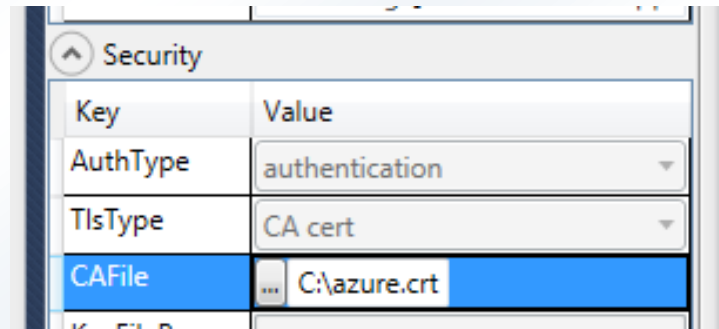
Al añadir este 'Gate' aparecerán los datos necesarios para configurar la comunicación con el dispositivo de Azure, como son el 'HostName', el nombre del dispositivo de Azure al que se quieren enviar sus datos y su clave de acceso principal:



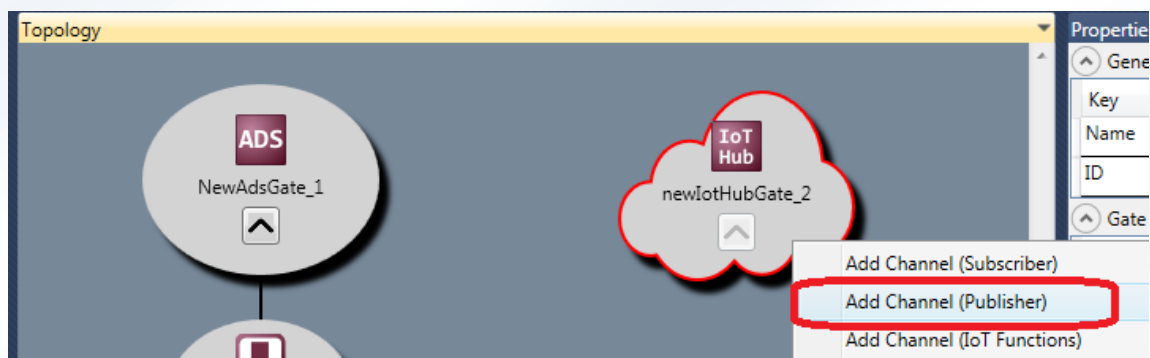
Estos datos se pueden obtener directamente del portal de Azure:



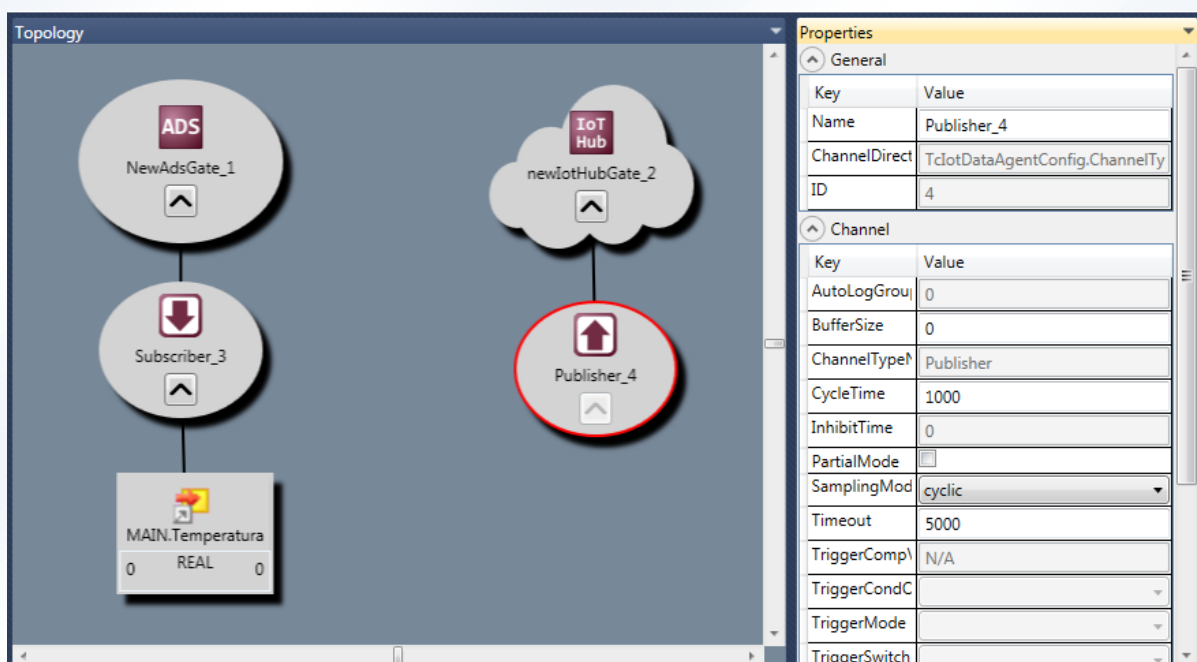
El ultimo campo, el 'CA file', es la ruta hacia el certificado para la conexión con Azure. Este fichero se puede generar como se ha explicado en el apartado ['Generar el certificado'](#) de este documento.



Sobre este 'Gate' hay que añadir un canal de tipo 'Publisher', para ello hay que pulsar con el botón derecho sobre él y seleccionar la opción de 'Add Channel ('Publisher')':



La configuración de este canal es prácticamente la misma que la configuración del canal 'Subscriber'. La configuración que hay que hacer es la frecuencia de envío de los datos (con las mismas opciones definidas para el 'Subscriber'). También se puede configurar el tamaño del buffer ante perdidas de conexión (por ejemplo, una caída de la conexión a internet) utilizando el campo de 'BufferSize'.



Por último, en el 'Publisher' se puede configurar el tipo de mensaje que se va a mandar (en el campo 'FormatterName'). Estos formatos pueden ser:

- **'RawBinary'**: se enviarán datos binarios sin procesar. Este formato obliga a que en el cliente en que se reciban los datos sea necesario decodificar estos datos siguiendo la estructura que tuviesen en TwinCAT.
- **'TwinCAT JSON'**: se enviará un mensaje JSON donde los valores estarán en un subcampo llamado 'Values' y se pueden añadir metadatos a las variables, como por ejemplo la unidad en que se manda la medida o su propio 'timestamp'. El siguiente mensaje muestra un ejemplo de esta configuración:

```
{
  "Timestamp": "2018-06-28T19:26:08.407+02:00",
  "GroupName": "Publisher",
  "Values":
  {
    "Humedad": 15.0,
    "Temperatura": 25.6
  },
  "MetaData":
  {
    "Humedad":
    {
      "Units": "°C"
    },
    "Temperatura":
    {
      "Units": "%"
    }
  }
}
```

- **'Simple JSON'**: envía un mensaje JSON estándar como se puede ver en el siguiente ejemplo:

```
{
  "Timestamp": "2018-06-28T19:16:59.151+02:00",
  "GroupName": "Publisher",
  "Humedad": 15.0,
  "Temperatura": 25.6
}
```

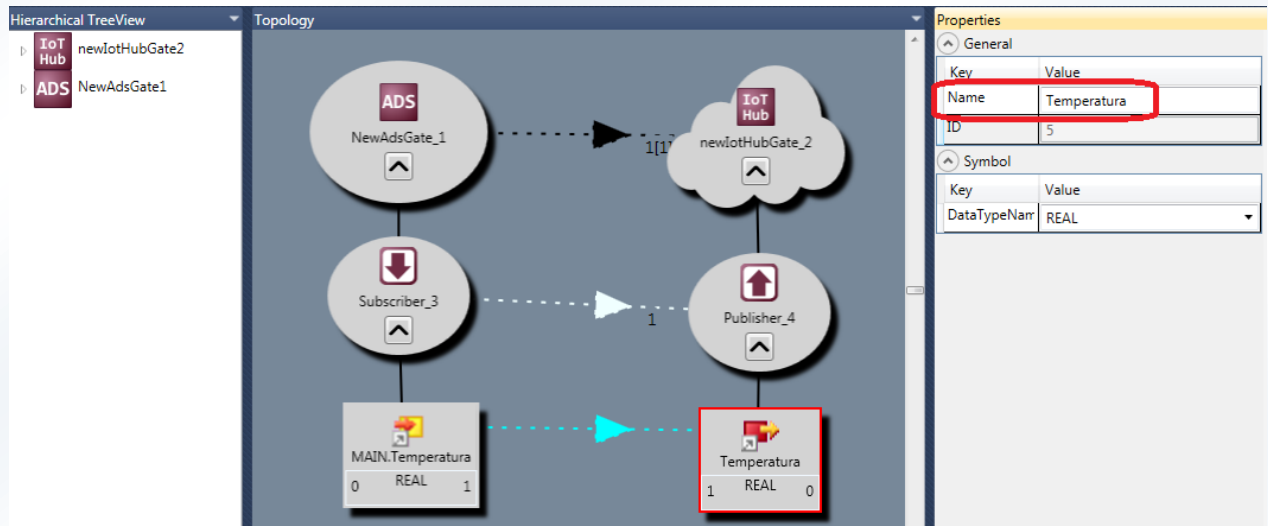
- **'TwinCAT Analytics'**: envía un mensaje formateado para que pueda ser interpretado por el paquete de TwinCAT Analytics correctamente.
- **'Text MW ThingSpeak'**: es el formato utilizado para enviar datos a la plataforma [ThingSpeak](#), que se suele utilizar para analizar los datos usando la herramienta de Matlab. Un ejemplo de este formato es el siguiente:

Temperatura=25.6&Humedad=15.0&created_at=2018-06-29T09:31:10.896

Una vez configurado el 'Publisher', el último paso que queda es crear las vinculaciones entre el canal de lectura de ADS y el de escritura en el dispositivo de Azure. Para ello hay que pulsar la tecla 'Ctrl' y hacer un drag&drop de la variable que se ha añadido en el 'Subscriber' llevándola sobre el bloque 'Publisher'.

Al hacer este proceso aparecerá una variable nueva sobre el bloque 'Publisher' vinculada a la variable leída por ADS.

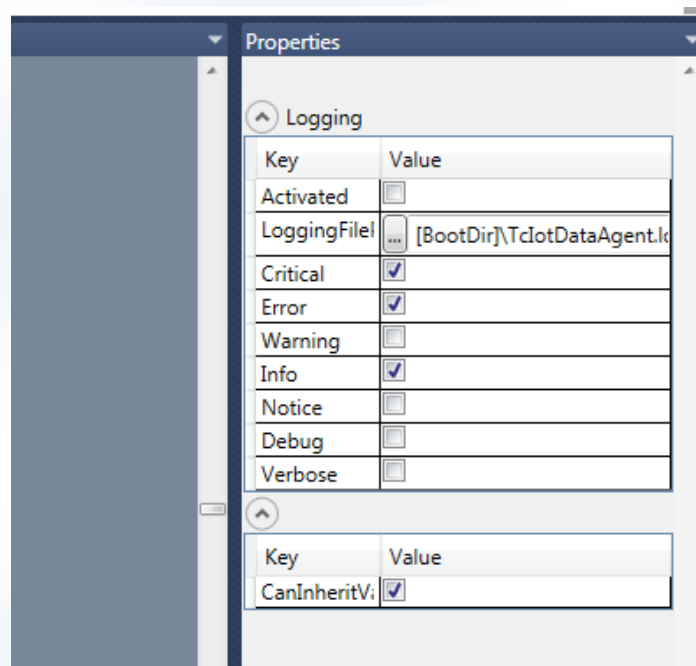
En las propiedades de esa nueva variable se puede modificar el campo 'Name' para cambiar el nombre de la variable en el mensaje JSON que se enviará a Azure. Hay que tener en cuenta que cuanto más largo sea el nombre más bytes ocupará el mensaje (más ancho de banda y un posible encarecimiento de la tarifa en caso de usar un modem 3G por ejemplo).



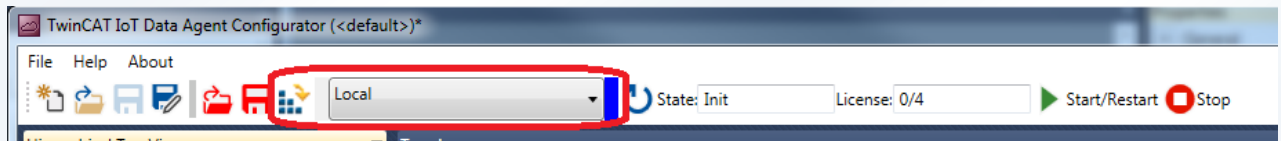
Hay que tener en cuenta que una variable de lectura se puede vincular a varios canales 'Publisher'. Del mismo modo, sobre un 'Gate' se pueden generar diferentes canales de lectura o escritura que trabajen a diferentes intervalos (por ejemplo, para leer señales analógicas de forma cíclica, señales digitales únicamente cuando cambien, etc...).

Estos mismos pasos habría que repetirlos con todos los 'Gates' y todas las variables que se quieran mandar.

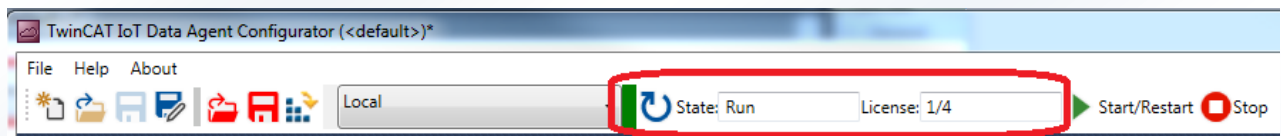
El Data Agent también es capaz de generar un log para poder registrar errores, avisos, etc... Para ello hay que pinchar en el fondo gris del esquema y en las propiedades aparecerá la opción de activar el log, donde guardar el fichero y los tipos de mensajes a registrar.



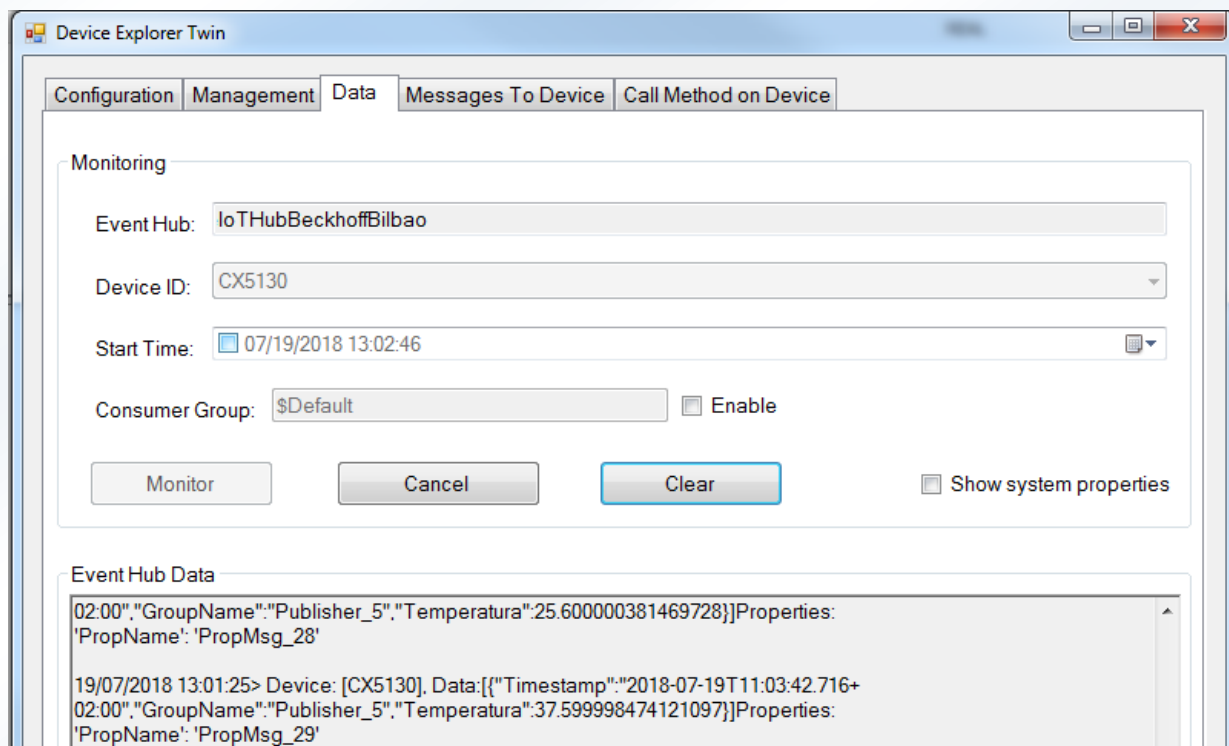
Una vez se tiene toda la configuración hay que seleccionar el equipo donde corre el Data Agent sobre el que se quiere descargar esta configuración y activarla:



Una vez descargada, si se pulsa sobre el botón para actualizar el estado se verá que ya está corriendo y se está consumiendo una licencia de las cuatro disponibles en el paquete básico del Data Agent (debido al 'Gate' de ADS). También se puede parar o reiniciar el servicio si fuese necesario.

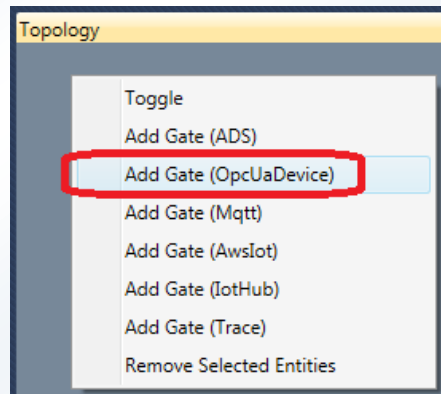


Una vez está corriendo el Data Agent, si se quieren monitorizar los mensajes que están llegando al centro de IoT Hub se puede hacer a través de la aplicación 'Device Explorer' en la pestaña 'Data'. En esta pestaña se puede seleccionar el dispositivo que se quiere monitorizar y al darle al botón de 'Monitor' se empezarán a recibir los mensajes:

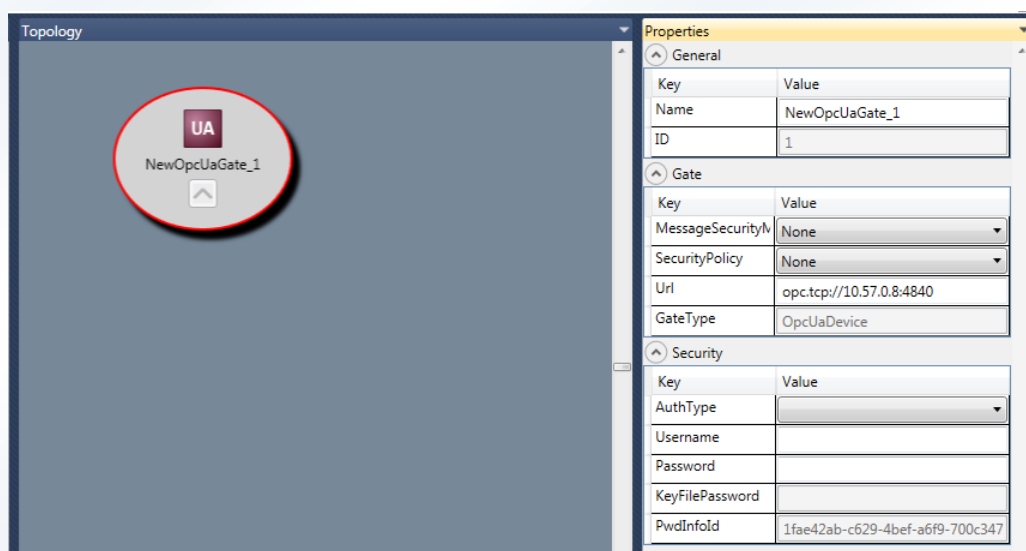


De la misma forma que se envían mensajes del dispositivo a la nube se podrían enviar mensajes de la nube al dispositivo. Los pasos de configuración serían los mismos, pero habría que añadir un canal de tipo 'Subscriber' en el 'Gate' del Azure IoT Hub y un canal de tipo 'Publisher' en el 'Gate' de ADS y vincular sus variables.

Del mismo modo se podrían leer datos de un dispositivo por OPC UA para subirlos a Azure. Lo único que habría que hacer es añadir un 'Gate' de tipo 'OpcUaDevice':

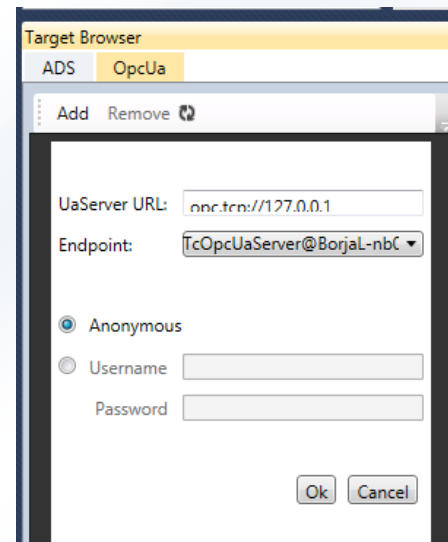
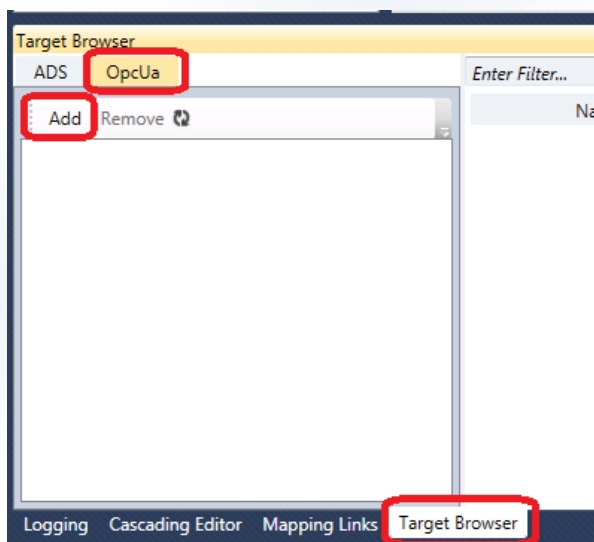


En este 'Gate' hay que poner la 'Url' del servidor OPC UA del que se quieren leer los datos y el tipo de seguridad de acceso en caso de que lo tenga configurado:



Después de ello hay que añadir el bloque 'Subscriber' igual que se ha explicado anteriormente con el 'Gate' de ADS.

Por último, para añadir las variables también está disponible el 'TargetBrowser' para el OPC UA. Para usarlo para explorar las variables, en primer lugar, hay que añadir el servidor OPC UA a la lista:



Tras añadirlo ya aparecerá el servidor listado y se podrá navegar por sus variables para poder arrastrarlas al bloque 'Subscriber':

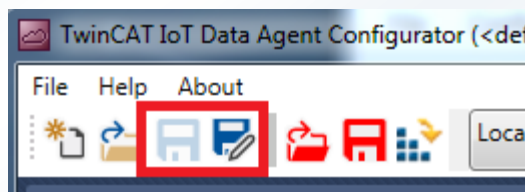
Target Browser

ADS OpcUa

Enter Filter...

| Name | Type | Size | Category | Full-Name | Comment | Subitems | NodeClass | Identifier | NamespaceIndex |
|----------------|--------|------|-----------|--------------|---------|----------|-----------|------------|----------------|
| Views | | 0 | Struct | Views | | 0 | Object | 87 | 0 |
| Objects | | 0 | Struct | Objects | | 5 | Object | 85 | 0 |
| Server | | 0 | Struct | Objects.S... | | 14 | Object | 2253 | 0 |
| DeviceSet | | 0 | Struct | Objects.D... | | 1 | Object | 5001 | 2 |
| PLC1 | | 0 | Struct | Objects.P... | | 14 | Object | PLC1 | 1 |
| DeviceManual | | 0 | Struct | Objects.P... | | 0 | Variable | PLC1.De | 1 |
| DeviceRevision | String | 4 | Primitive | Objects.P... | | 0 | Variable | PLC1.De | 1 |
| HardwareRevisi | String | 4 | Primitive | Objects.P... | | 0 | Variable | PLC1.Ha | 1 |
| Manufacturer | | 0 | Struct | Objects.P... | | 0 | Variable | PLC1.Ma | 1 |

Si se quiere guardar el proyecto de configuración del Data Agent se puede exportar a un fichero '.xml' con la siguiente opción:



Si se abre este fichero '.xml' con un editor de texto se puede ver el contenido del fichero y también se podría modificar manualmente si fuese necesario.

```

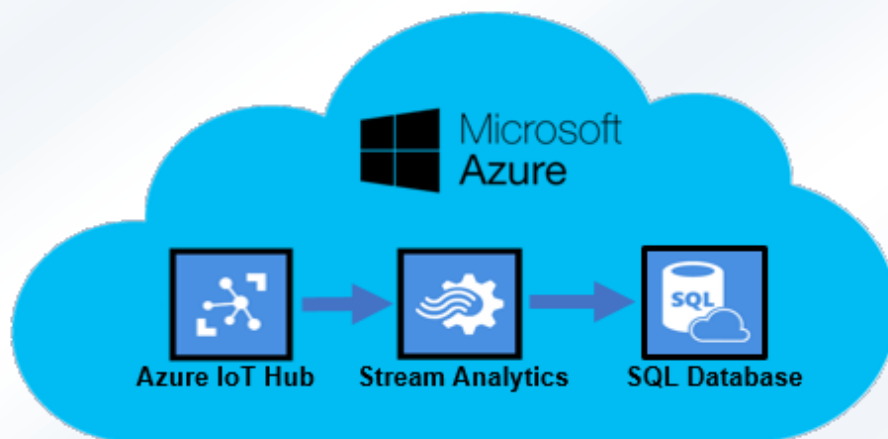
1 <?xml version="1.0" encoding="utf-8"?>
2 <Configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:noNamespaceSchemaLocation="ConfigSchema_IoT_Version_4.0.1.0.xsd">
3   <Header>
4     <Timestamp>2018-06-29T07:27:45Z</Timestamp>
5   </Header>
6   <GlobalConfig>
7     <BaseConfig>
8       <FormatterInstances />
9       <FormatterTypes />
10      <Gates>
11        <Gate Id="1" ReferenceTo="0" Switch="ON" Type="AdsClient" IoMode="batched" Name="NewAdsGate_1">
12          <NetId>192.168.0.19.1.1</NetId>
13          <Port>851</Port>
14          <VisualData>
15            <AuthenticationSettings PwdInford="e77b5e9b-8448-4f30-b153-fdea875a2686" />
16            <Channels>
17              <Channel Id="1" ReferenceTo="0" Name="Subscriber_1" Switch="ON" Role="Subscriber" SamplingMode="cyclic" CycleTime="1000" Timeout="5000" Retain="false" MappingId="0" InhibitTime="0" BufferSize="0" OnChangePartialMode="true">
18                <VisualData>
19                  <X>98.19999999999999</X>
20                  <Y>196.04000000000002</Y>
21                  <Width>120</Width>
22                  <Height>100</Height>
23                  <FontSize>12</FontSize>
24                  <IsRenderable>true</IsRenderable>
25                </VisualData>
26                <Formatter EnableSymbolTimestamp="true" ProgId="{EC125306-74D0-46E4-9518-9258DB043679},ToDataAgent.Formatter.Binary.1" Name="Raw Binary" Type="InOut" />
27                <Symbol Id="1" ReferenceTo="0" Switch="ON" DataType="REAL" DataTypeGuid="{18071995-0000-0000-0000-000000000000}" IdxGroup="16448" IdxOffset="512028" BitSize="16" />
28                <VisualData>
29                  <X>108</X>
30                  <Y>348.03999999999996</Y>
31                </VisualData>
32              </Channel>
33            </Channels>
34          </Gate>
35        </Gates>
36      </BaseConfig>
37    </GlobalConfig>
38  </Configuration>

```

5. GUARDAR DATOS EN AZURE SQL

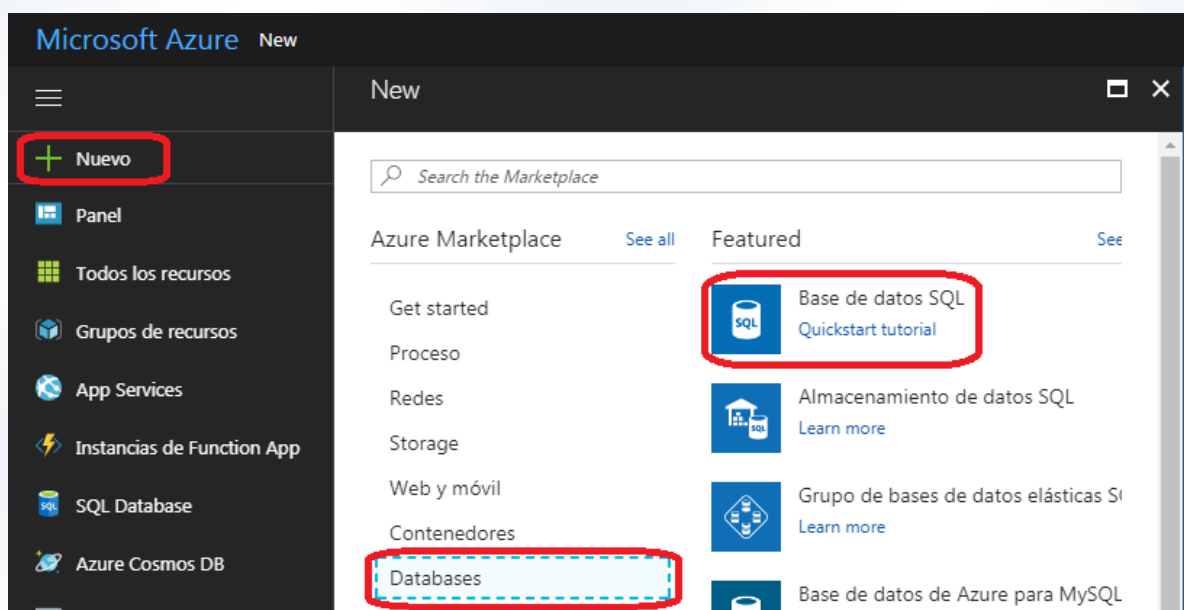
En este apartado se va a realizar un ejemplo de guardado de los datos que se suben desde una CPU con TwinCAT a un dispositivo de IoT Hub en una base de datos SQL sobre Azure, lo que permitirá una mayor seguridad de acceso que una base de datos normal y ofrece muchas posibilidades a nivel de escalabilidad ([enlace de información](#)).

Para realizar esta arquitectura se hace uso de la herramienta del objeto Stream Analytics Job de la suite de Azure ([enlace de información](#)). Este objeto será una función SQL que se ejecutará cada vez que se reciba un dato en el IoT Hub y lo guardará sobre la base de datos.

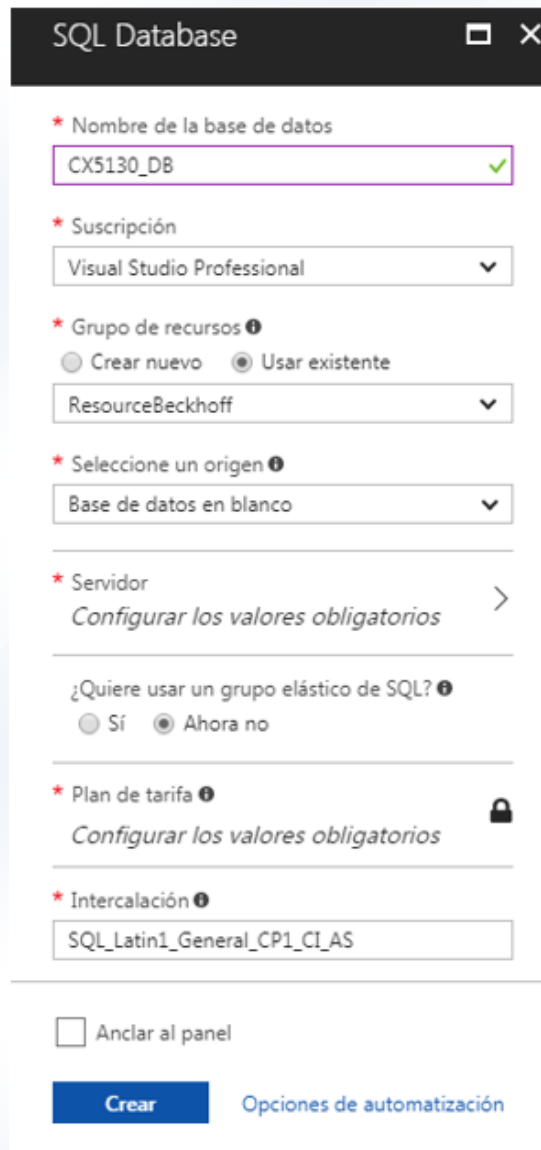


CONFIGURAR LA BASE DE DATOS DE SQL EN AZURE

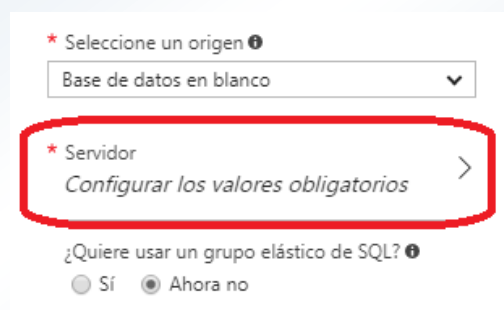
Para crear la base de datos sobre la suite de Azure hay que utilizar la siguiente opción desde el portal de Azure:



Entre los parámetros que hay que configurar están el nombre de la base de datos, el recurso a utilizar (en este caso para aumentar la velocidad se utilizará el mismo grupo de recursos que el usado para el IoT Hub), los datos del servidor y la tarifa a aplicar.



Para configurar los datos del servidor hay que pinchar sobre la siguiente opción:



En los parámetros del servidor hay que definir la URL con la que se accederá a la base de datos (especificando el nombre del servidor) y configurar un usuario y un password para poder acceder a sus datos.

The screenshot shows three panels in the Azure portal:

- SQL Database:**
 - Nombre de la base de datos: CX5130_DB
 - Suscripción: Visual Studio Professional
 - Grupo de recursos: ResourceBeckhoff
 - Selección un origen: Base de datos en blanco
 - Servidor: Configurar los valores obligatorios
 - ¿Quiere usar un grupo elástico de SQL?: Sí
 - Plan de tarifa: Configurar los valores obligatorios
 - Intercalación: SQL_Latin1_General_CP1_CI_AS
 - Anclar al panel: ☒
 - Crear
 - Opciones de automatización
- Servidor:**
 - Crear un servidor nuevo
 - No se han encontrado servidores
- Nuevo servidor:**
 - Nombre del servidor: cx5130dbserver
 - Inicio de sesión del administrador del servidor: BeckhoffBilbao
 - Contraseña: [oculto]
 - Confirmar contraseña: [oculto]
 - Ubicación: Oeste de Europa
 - Permitir que los servicios de Azure accedan al servidor: ☒
 - Seleccionar

Por último, en la configuración de la base de datos habrá que seleccionar la tarifa a utilizar (que limitará el tamaño que podrá tener la base de datos).

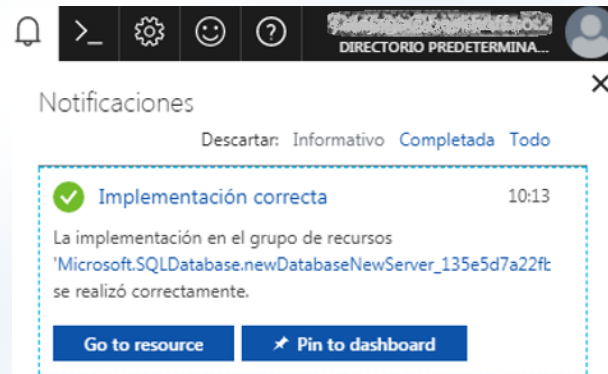
The close-up shows the 'Plan de tarifa' section with the text 'Configurar los valores obligatorios' and a lock icon.

Una vez configurado todo, debería quedar algo similar a la siguiente imagen:

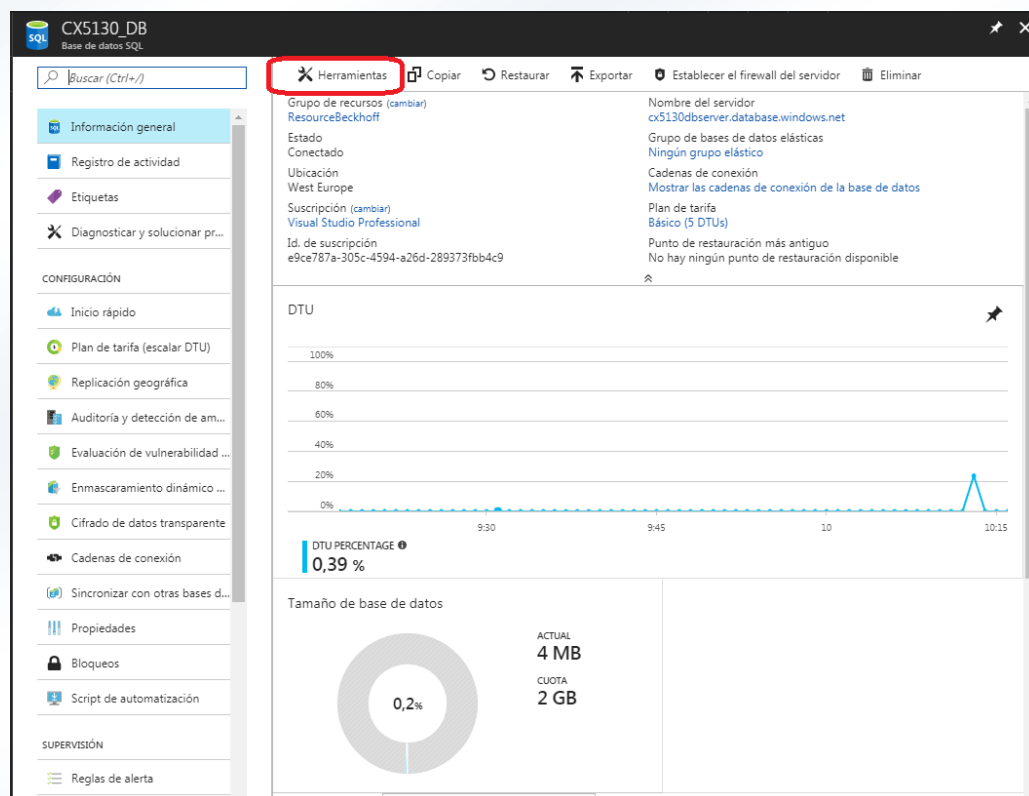
The screenshot shows the final configuration of the SQL Database:

- Nombre de la base de datos: CX5130_DB
- Suscripción: Visual Studio Professional
- Grupo de recursos: ResourceBeckhoff
- Selección un origen: Base de datos en blanco
- Servidor: cx5130dbserver (Oeste de Europa)
- ¿Quiere usar un grupo elástico de SQL?: Ahora no
- Plan de tarifa: Basic: 5 DTU, 2 GB
- Intercalación: SQL_Latin1_General_CP1_CI_AS
- Anclar al panel: ☒
- Crear
- Opciones de automatización

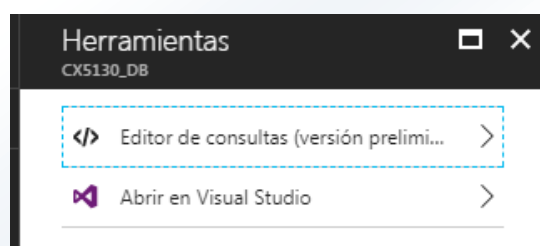
Una vez comprobados todos los datos hay que darle al botón de 'Crear' y esperar a la notificación que indique que se ha creado correctamente (puede tardar unos minutos):



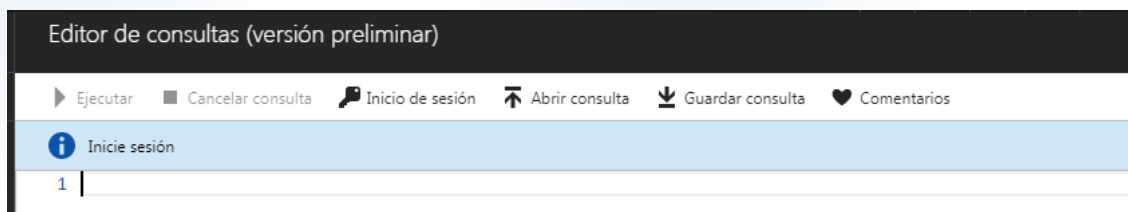
Una vez creada la base de datos, el siguiente paso es crear una tabla donde quedarán alojados los datos. Para ello, desde el panel de control se puede pinchar sobre la base de datos recién creada y en la sección de 'Información general' estará disponible el botón de 'Herramientas':



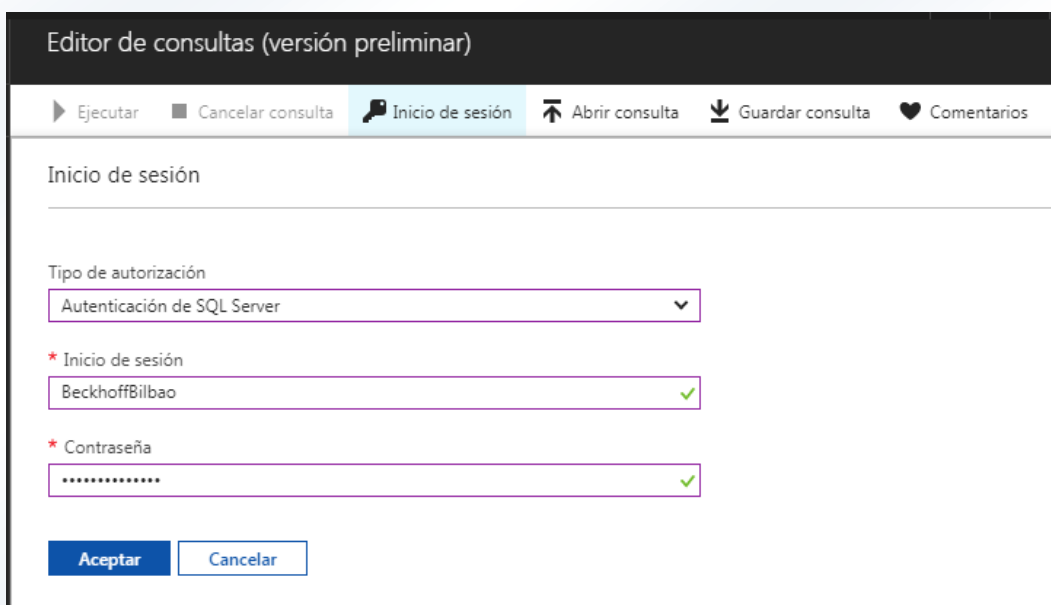
En estas herramientas hay un editor de consultas SQL desde el portal de Azure (en el momento de hacer la guía estaba en versión preliminar) y la opción de abrir el editor de consultas SQL propio de Visual Studio. En este caso se va a utilizar el portal de Azure, pero más adelante se puede ver como configurar la conexión desde el cliente SQL de Visual Studio.



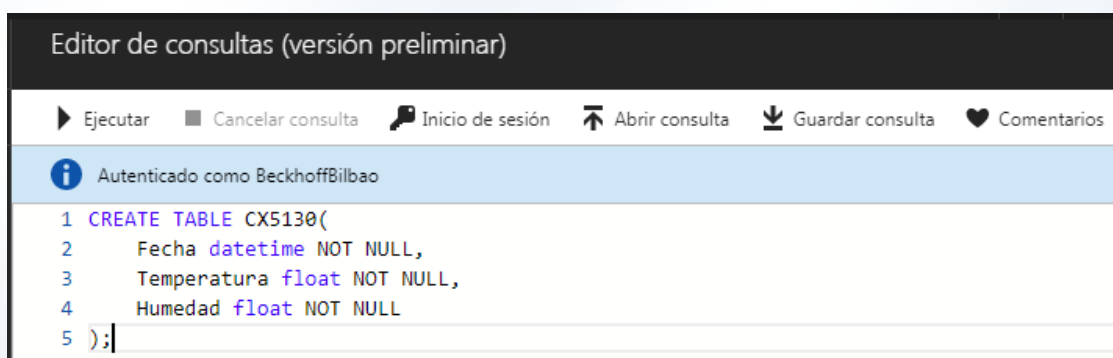
Una vez abierto el editor de consultas se podrá escribir sobre el cualquier query en lenguaje SQL y ejecutarlo:



Como se puede ver, la opción de 'Ejecutar' esta deshabilitada debido a que aún no se ha iniciado sesión en la base de datos, para ello hay que pinchar en la opción de 'Inicio de sesión' e introducir los datos de inicio de sesión definidos en el momento en que se creó la base de datos.



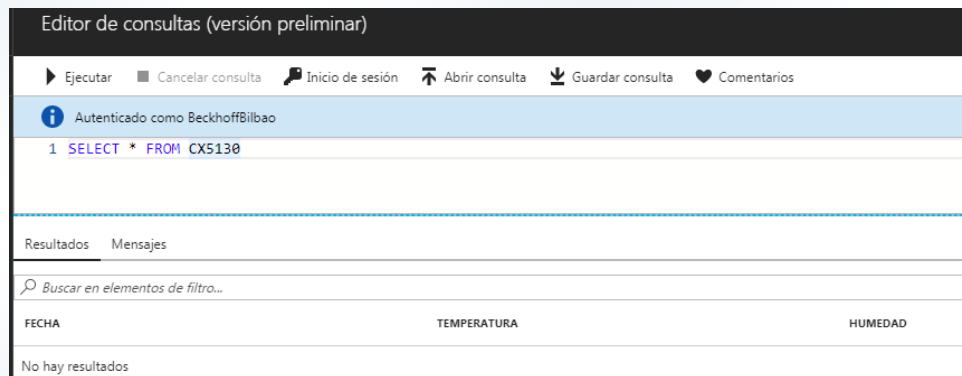
Una vez se ha iniciado sesión se va a crear la tabla que guardará los datos. En este ejemplo la tabla tendrá un campo con la fecha de recogida del dato y los valores de temperatura y humedad medidos (los valores que se subirán desde TwinCAT).



Al darle a ejecutar, si todo se ha ejecutado correctamente, debería salir el siguiente mensaje:

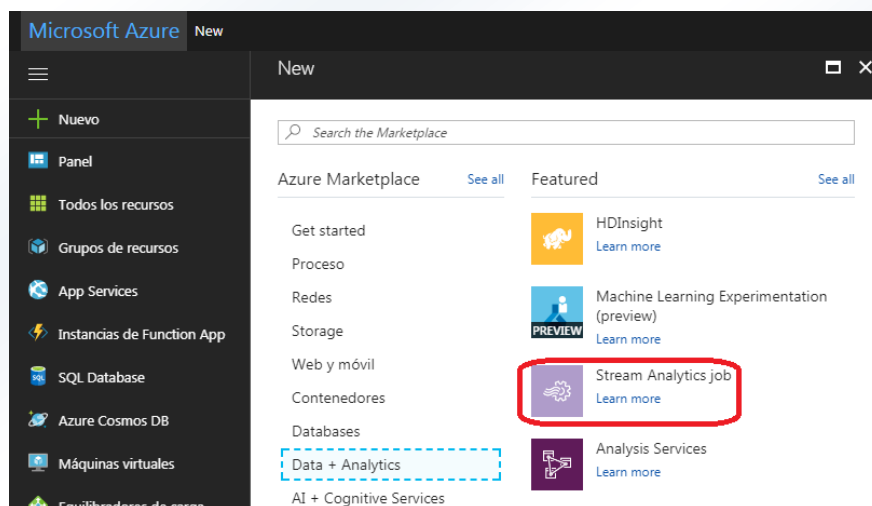


Para confirmar que se ha creado la tabla correctamente se puede hacer una consulta sobre ella (aunque de momento no hay datos guardados y solo se verá el encabezado).

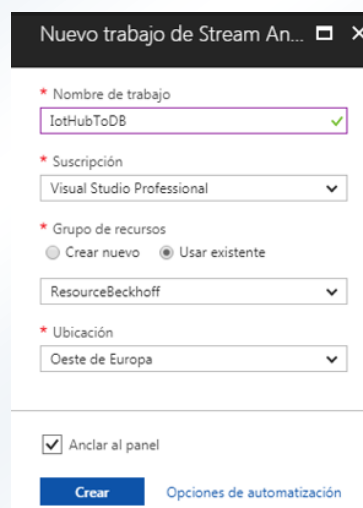


CONFIGURAR EL STREAM ANALYTICS JOB

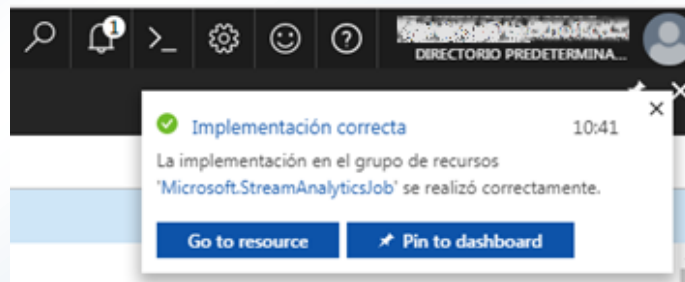
El siguiente paso es crear la rutina del Stream Analytics Job que se encargará de leer los datos recibidos del IoT Hub y los volcará a la base de datos. Para crear este objeto hay que seleccionar la siguiente opción desde el portal de Azure.



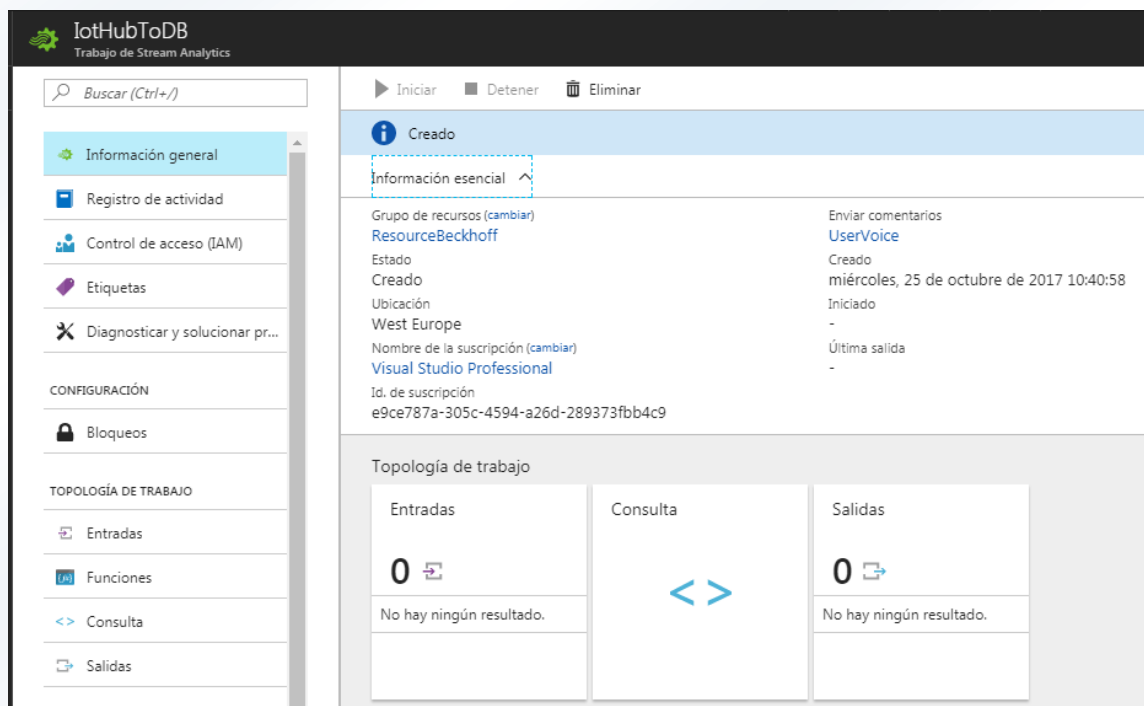
En la configuración de este objeto se le dará un nombre y se seleccionará el grupo de recursos a utilizar (en este caso se hospedará en el mismo que la base de datos y el centro de IoT Hub).



Al pulsar sobre el botón de 'Crear' habrá que esperar a la notificación de que se ha creado correctamente:

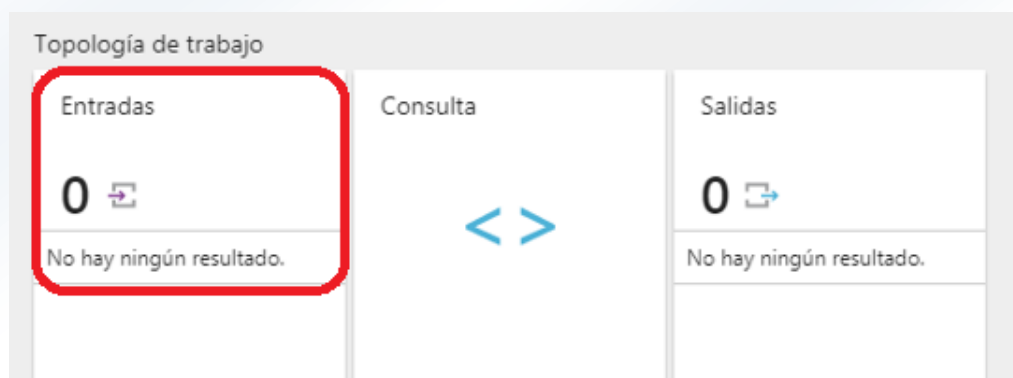


Una vez se ha creado se puede pinchar sobre él desde el panel de control para ir al panel de configuración del Stream Analytics Job.



En este panel se puede arrancar y detener el trabajo y se pueden definir los datos de entrada y salida que usará el trabajo y la función que realizará.

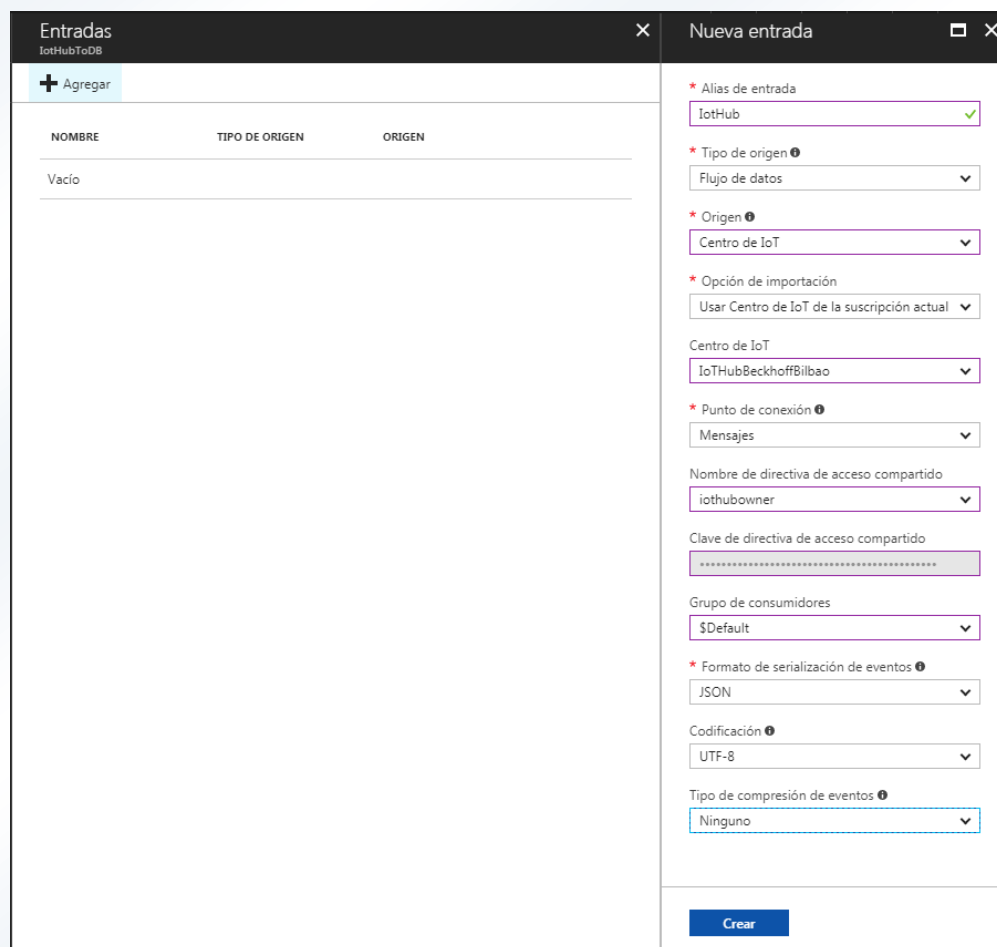
En primer lugar hay que pulsar sobre el campo de 'Entradas' dentro de la topología para definir la fuente de entrada de datos:



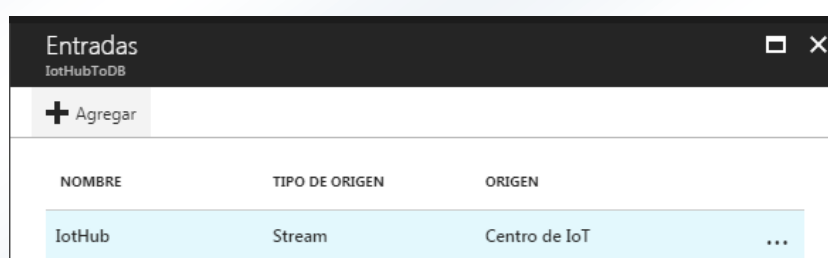
Dentro de la configuración de las entradas hay que pulsar sobre la opción de 'Agregar' para añadir una entrada.



En la configuración de la entrada hay que seleccionar el origen de los datos, en este caso hay que seleccionar el 'Centro de IoT'. En caso de que solo haya un centro de IoT creado el resto de datos se rellenarán automáticamente, sino habría que seleccionar el centro del que se quieren recibir los datos para guardarlos en la base de datos entre todos los disponibles.



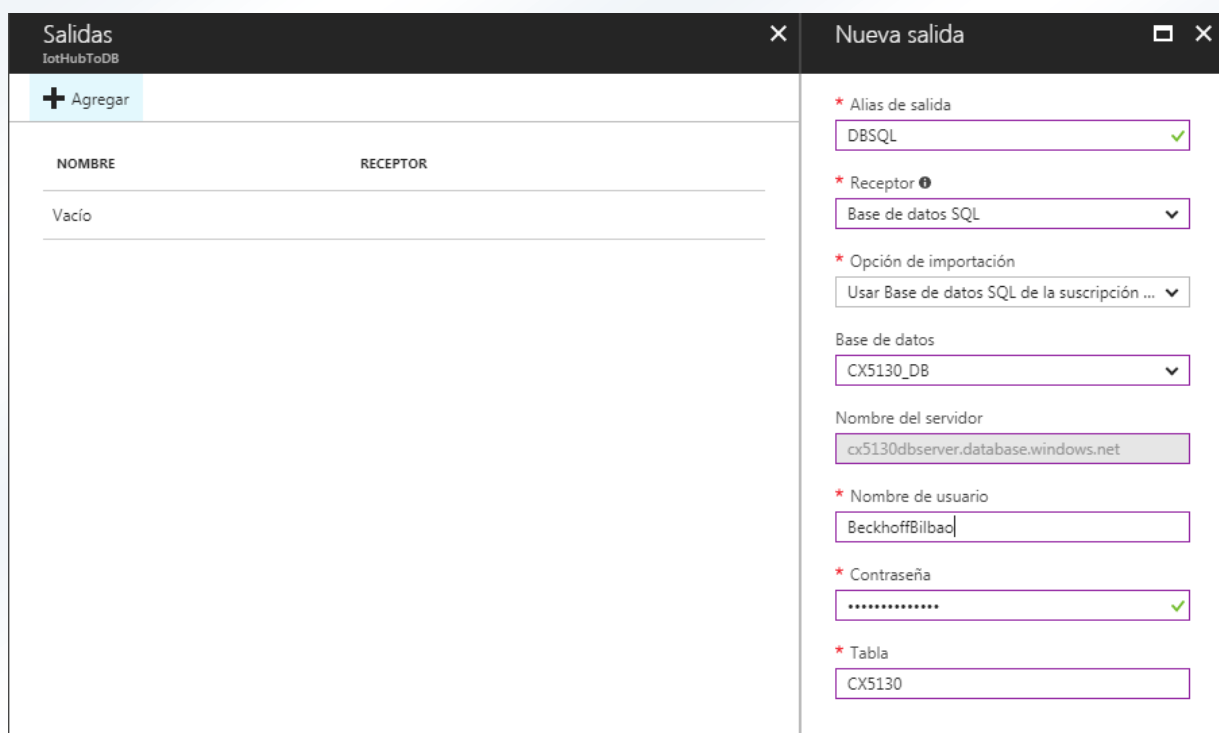
Al darle a la opción de crear se mostrará este origen de datos en la lista de entradas del Stream Analytics Job.



El siguiente paso será definir la salida del Stream Analytics Job, para ello hay que pulsar sobre las salidas en la topología del trabajo. Una vez en el listado de las salidas hay que darle a la opción de 'Agregar' para añadir una nueva salida.



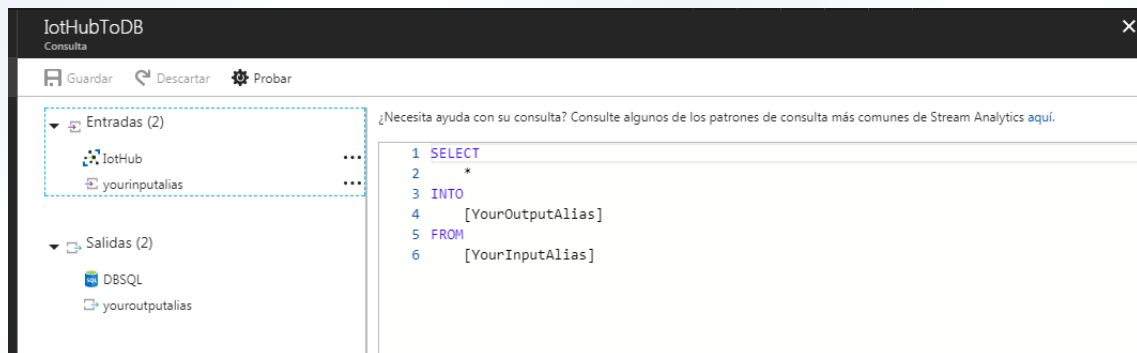
En la configuración de las salidas hay que seleccionar como salida la base de datos SQL creada en el apartado anterior. Además, habrá que configurar el usuario, el password y la tabla donde se van a guardar los datos.



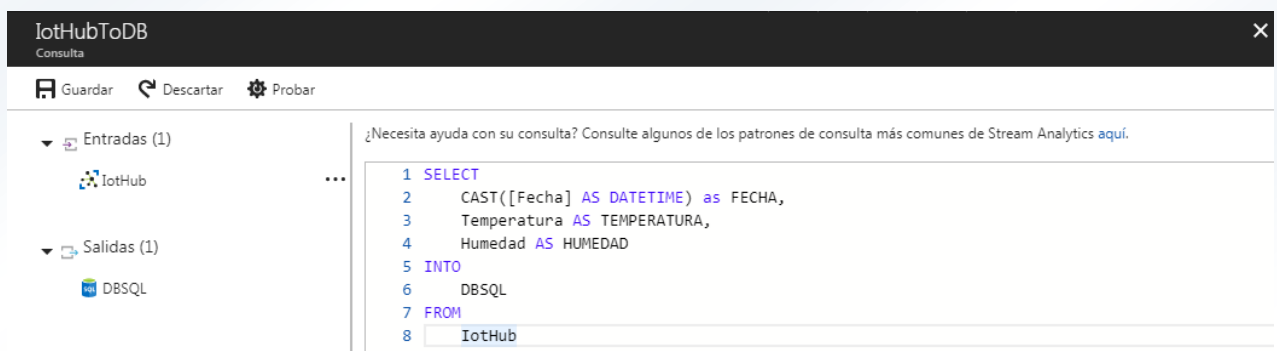
Al darle a la opción de crear se mostrará esta salida de datos en la lista de salidas del Stream Analytics Job.



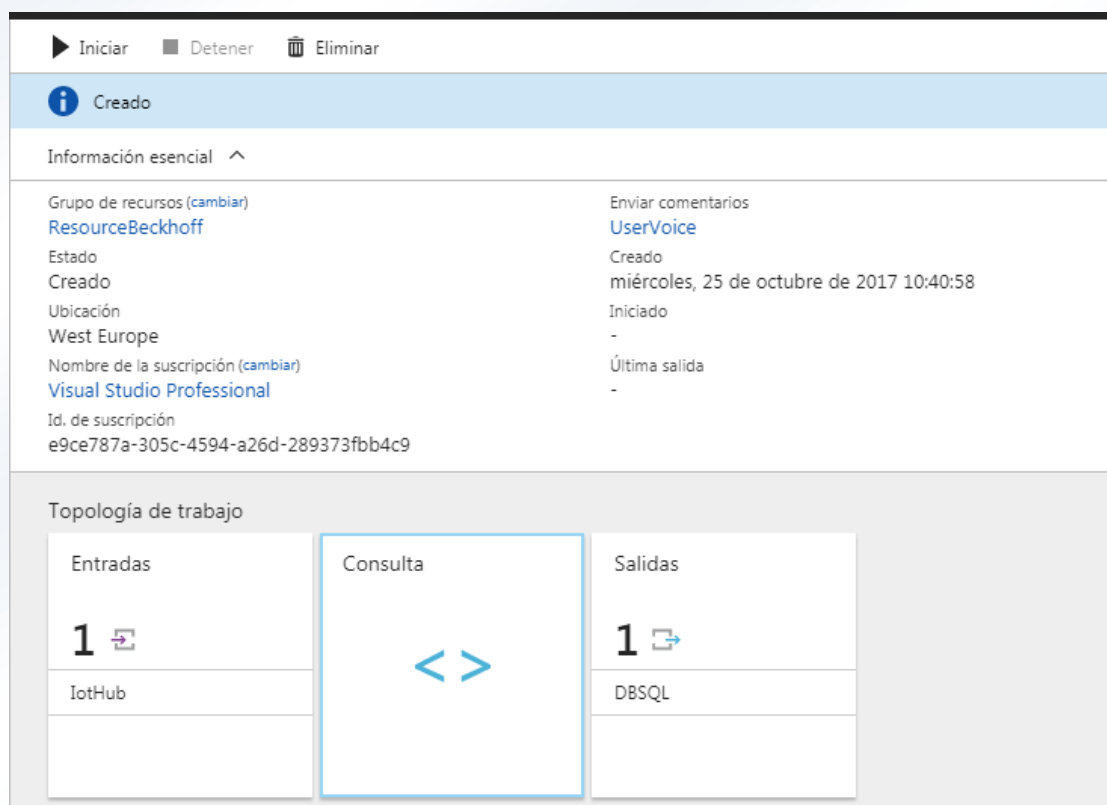
El último paso será definir la consulta SQL del trabajo. Al pinchar sobre las consultas para crearla aparecerá el siguiente código base:



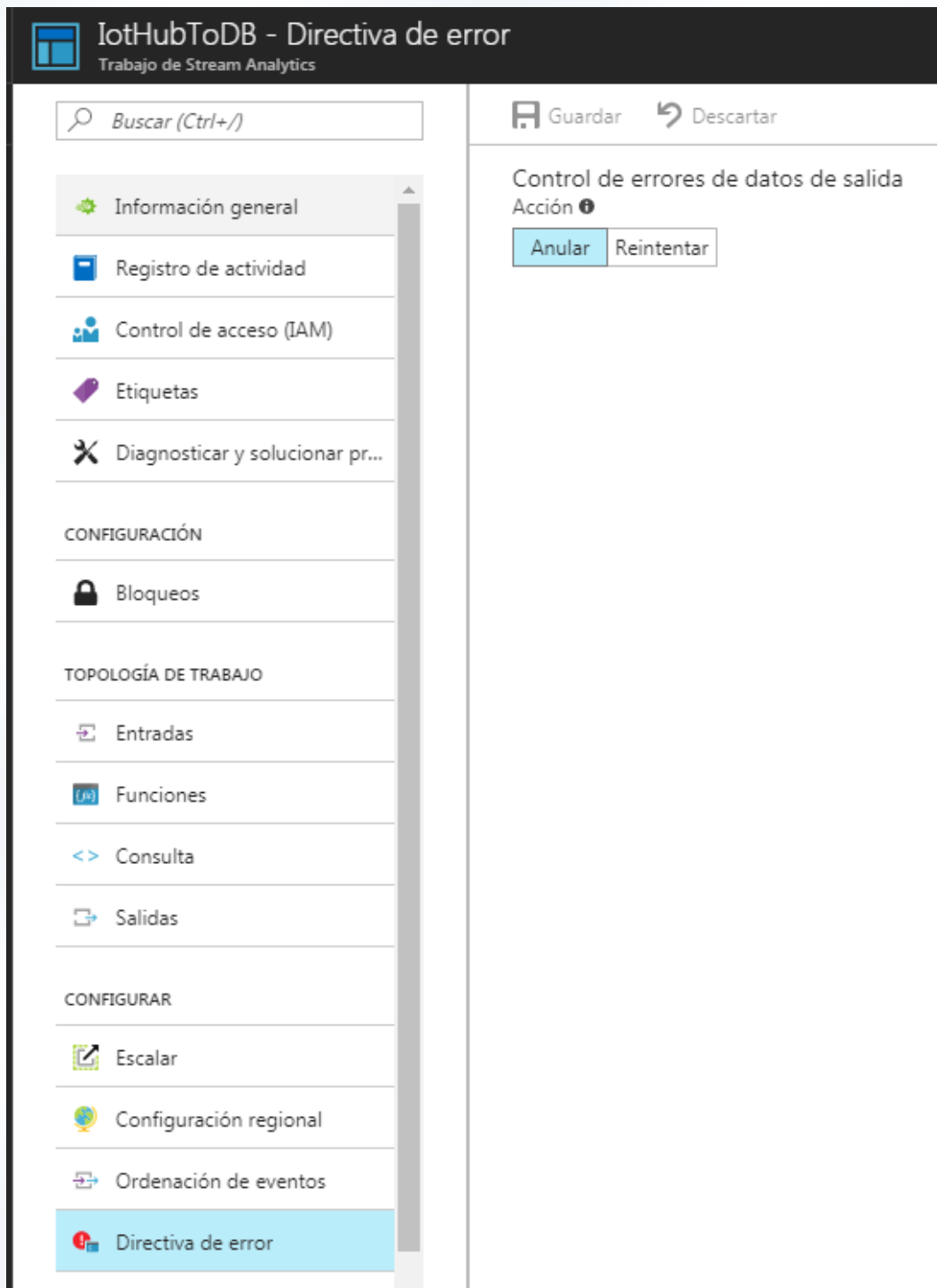
El código que hay que añadir es el de la siguiente imagen. En él se cogen los campos 'Fecha', 'Temperatura' y 'Humedad' del mensaje JSON que se recibirá en el centro de datos (convirtiendo la 'Fecha' a un objeto de tipo DATETIME) y se escriben sobre la base de datos.



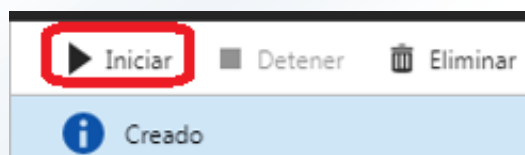
Una vez guardada la consulta se puede ver en el panel de administración del Stream Analytics Job todos los parámetros configurados.



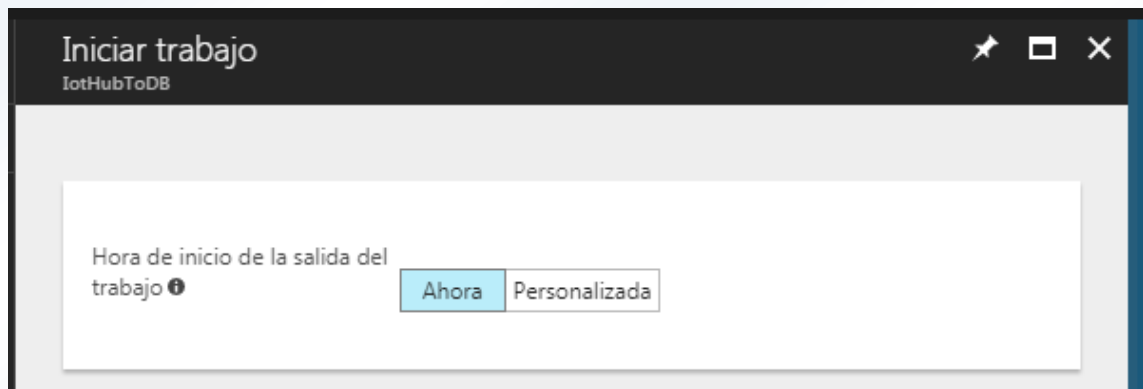
Hay que tener en cuenta que si en el telegrama no se recibiese alguno de los campos definidos en la consulta dentro del JSON (por ejemplo, no se recibe el dato de 'Temperatura') o en un formato incorrecto, el trabajo entraría en error y dejaría de ejecutarse. Para evitar esto se puede seleccionar la opción de 'Anular' dentro del menú de 'Directiva de error' y darle a 'Guardar':



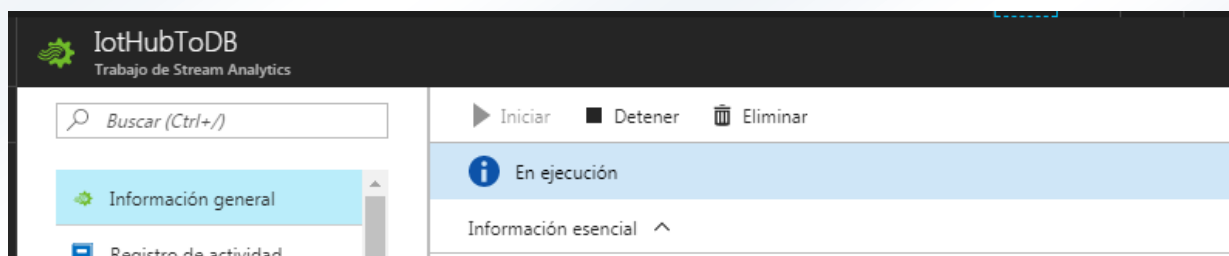
Una vez esté todo configurado hay que darle al botón de 'Iniciar':



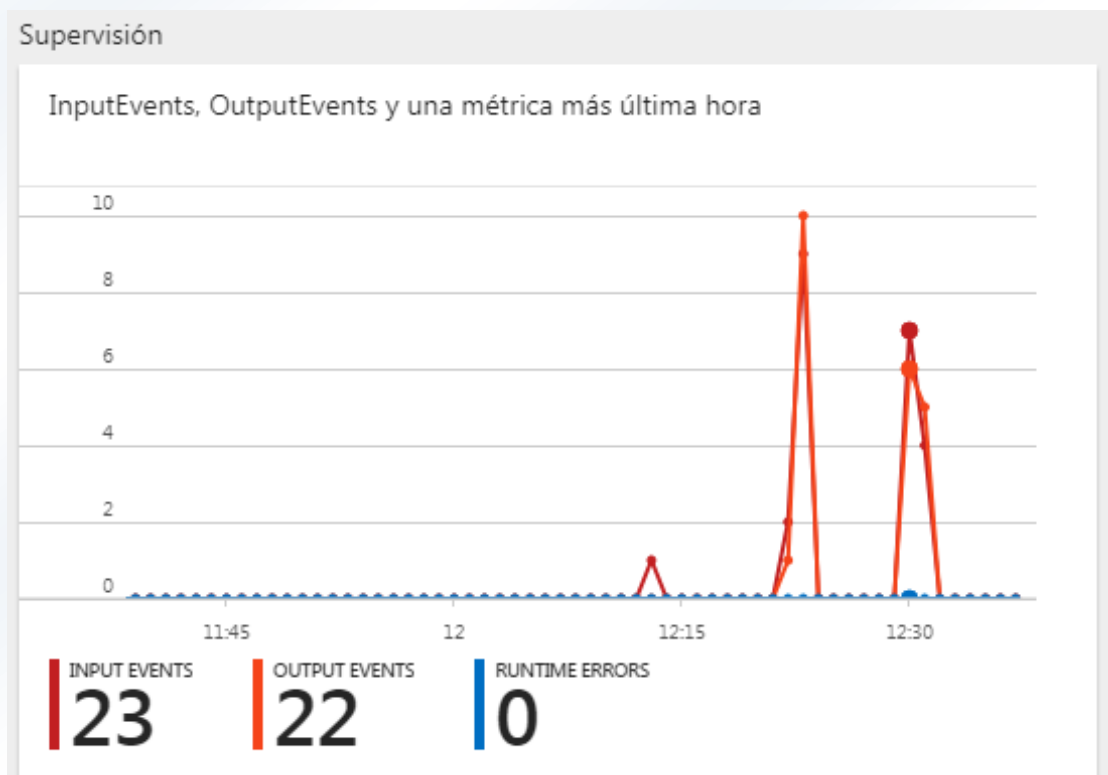
Si fuese necesario se podría aplazar la hora para iniciar el trabajo, en este caso se seleccionará la opción 'Ahora' para que se inicie inmediatamente:



Una vez se haya iniciado el trabajo se puede comprobar su estado dentro de la 'Información general' del Stream Analytics Job.



En esta misma sección también se puede comprobar el número de eventos de entrada (mensajes recibidos al IoT Hub), eventos de salida (escritura en la base de datos SQL) y errores producidos.



VISUALIZAR LOS DATOS GUARDADOS EN LA BASE DE DATOS DESDE EL PORTAL DE AZURE

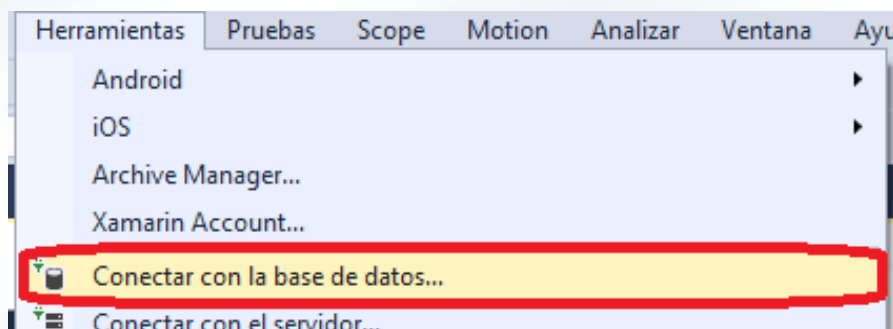
Desde el editor de consultas del portal de Azure se puede hacer una consulta SQL para leer todos los datos de la tabla:

| Editor de consultas (versión preliminar) | | |
|---|------------------|--------------------|
| ▶ Ejecutar ■ Cancelar consulta 🗨 Inicio de sesión ⬆ Abrir consulta ⬇ Guardar consulta ❤ Comentarios | | |
| i Autenticado como BeckhoffBilbao | | |
| 1 SELECT * FROM CX5130 | | |
| Resultados Mensajes | | |
| 🔍 Buscar en elementos de filtro... | | |
| FECHA | TEMPERATURA | HUMEDAD |
| 2017-10-25T12:38:01.0930000 | 22.9053173065186 | 4.79301118850708 |
| 2017-10-25T12:38:06.1900000 | 76.8108673095703 | 5.98062086105347 |
| 2017-10-25T12:38:11.2900000 | 43.486686706543 | 6.65156030654907 |
| 2017-10-25T12:38:16.3900000 | 14.8058290481567 | 0.0460822470486164 |
| 2017-10-25T12:38:21.4870000 | 84.493782043457 | 5.04493761062622 |
| 2017-10-25T12:38:26.5870000 | 27.1335926055908 | 7.82970952987671 |
| 2017-10-25T12:38:31.6870000 | 68.7525787353516 | 7.02876329421997 |
| 2017-10-25T12:38:36.7830000 | 96.0067138671875 | 3.24193167686462 |
| 2017-10-25T12:38:41.8830000 | 4.71198606491089 | 5.28297853469849 |
| 2017-10-25T12:38:46.9800000 | 19.6520938873291 | 8.97566223144531 |
| 2017-10-25T12:38:52.0800000 | 58.0971984863281 | 9.54009342193604 |

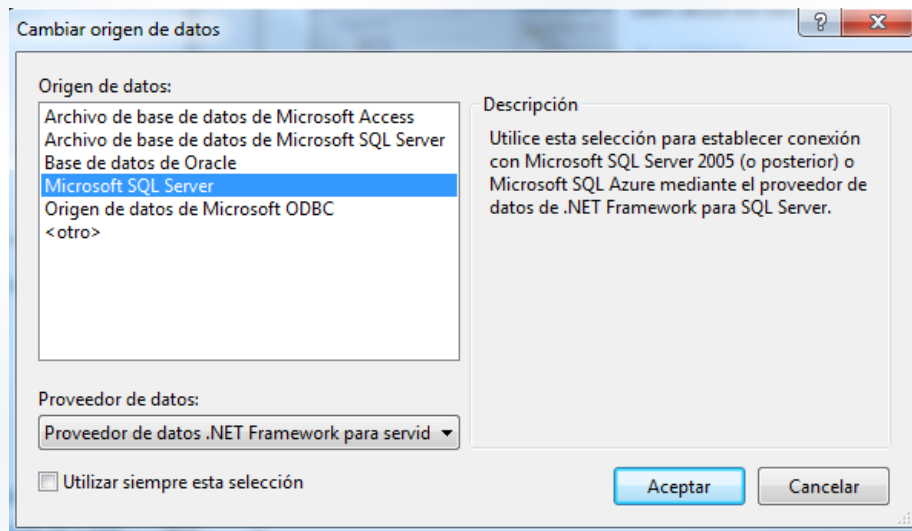
VISUALIZAR LOS DATOS GUARDADOS EN LA BASE DE DATOS DESDE VISUAL STUDIO

Al ser una base de datos de SQL, aunque esté hospedada en la suite de Microsoft Azure, se pueden acceder a los datos desde otros clientes SQL. En este caso se va a mostrar el proceso para leer los datos desde la herramienta de conexiones de bases de datos de Visual Studio.

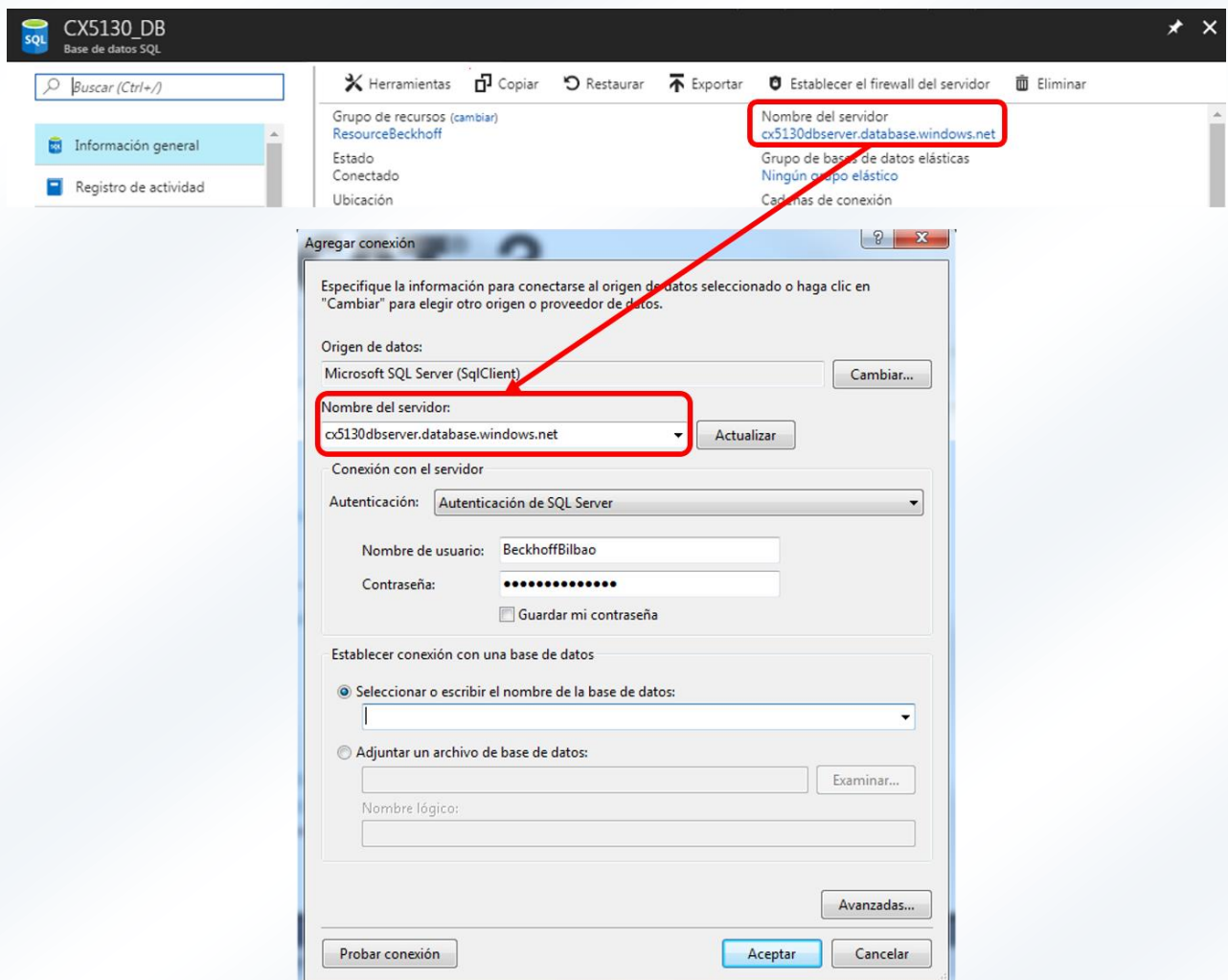
Para abrir esta herramienta hay que ir a la sección 'Herramientas' y elegir la opción 'Conectar con la base de datos'.



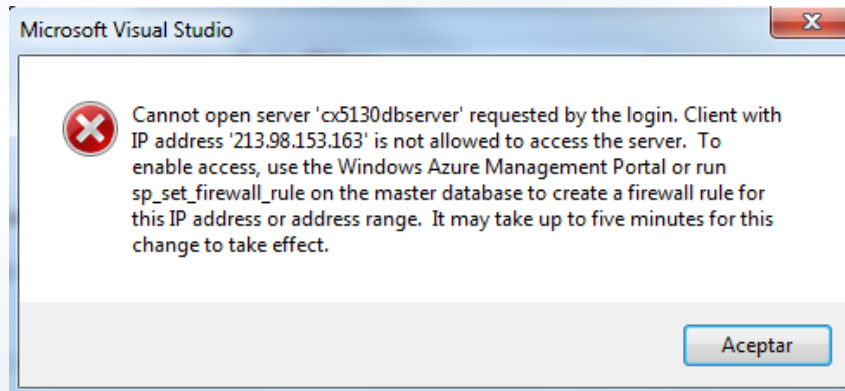
Dentro de las opciones de origen de datos del asistente hay que seleccionar la opción 'Microsoft SQL Server':



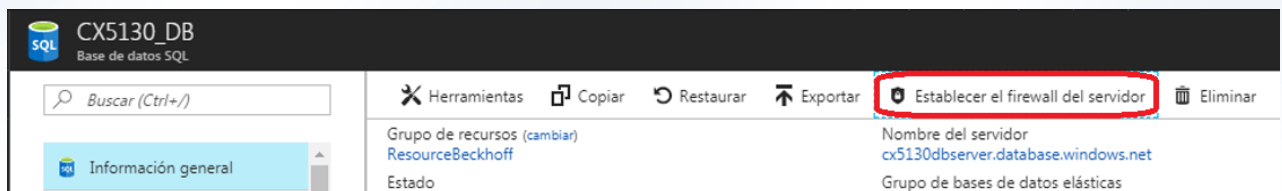
En el asistente para agregar la conexión hay que escribir el nombre del servidor de la base de datos de Azure. Este nombre se puede ver en el portal de Azure. También hay que escribir el nombre de usuario y el password de acceso:



En este punto si se intenta seleccionar el nombre de la base de datos desde el seleccionable saldrá el siguiente mensaje de error (la dirección IP cambiará mostrando la IP publica del PC desde el que se está trabajando):



Este error se debe a que, por motivos de seguridad, en Azure es necesario añadir al firewall de la base de datos las IPs de los clientes que se van a conectar a ella. Para ello, desde el portal de Azure, en la administración de la base de datos hay que seleccionar la opción 'Establecer el firewall del servidor':




Dentro de la configuración del firewall se pueden añadir excepciones a IPs que sí podrán conectarse a esta base de datos.



Si se pulsa al botón de 'Agregar IP de cliente' se añadirá como regla la IP desde la que se ha accedido al portal de Azure.

Guardar
Descartar
+ Agregar IP de cliente


Las conexiones de las IP especificadas a continuación proporcionan acceso a todas las bases de datos en cx5130dbserver.

Permitir el acceso a servicios de Azure

ACTIVADO
DESACTIVADO

Dirección IP de cliente
213.98.153.163

| NOMBRE DE REGLA | IP INICIAL | IP FINAL | |
|------------------------------|----------------|----------------|-----|
| | | | ... |
| ClientIPAddress_2017-10-2... | 213.98.153.163 | 213.98.153.163 | ... |

Hay que tener en cuenta que si el equipo que se va a conectar habitualmente a la base de datos SQL está configurado con IP dinámica habría que añadir el rango de IPs que asigna el servidor de internet a ese PC (consultando con el servidor de internet ese rango).

Una vez configurada la IP en el firewall, de vuelta en la herramienta de acceso a bases de datos de Visual Studio, hay que cerrar y volver a abrir el dialogo y ya se podría conectar con el servidor sin errores. En este punto hay que seleccionar el nombre de la base de datos en el desplegable.

Agregar conexión

Especifique la información para conectarse al origen de datos seleccionado o haga clic en "Cambiar" para elegir otro origen o proveedor de datos.

Origen de datos:

Microsoft SQL Server (SqlClient)
Cambiar...

Nombre del servidor:

cx5130dbserver.database.windows.net
Actualizar

Conexión con el servidor

Autenticación:

Autenticación de SQL Server

Nombre de usuario:

BeckhoffBilbao

Contraseña:

.....

☐ Guardar mi contraseña

Establecer conexión con una base de datos

☒ Seleccionar o escribir el nombre de la base de datos:

CX5130_DB

☐ Adjuntar un archivo de base de datos:

Examinar...

Nombre lógico:

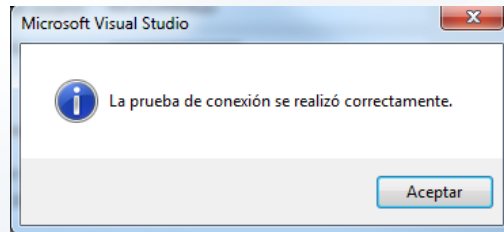
Avanzadas...

Probar conexión

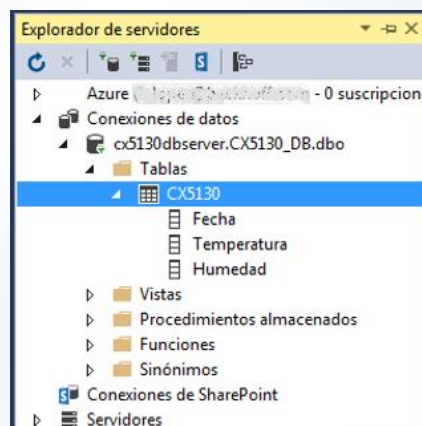
Aceptar

Cancelar

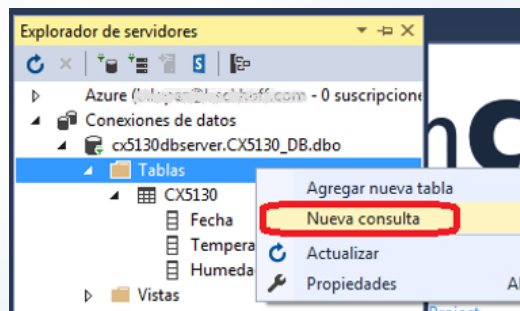
Una vez añadidos todos los datos hay que pulsar sobre el botón 'Probar conexión' para comprobar que todo está configurado correctamente. En caso de que así sea deberá aparecer el siguiente mensaje:



Al darle a 'Aceptar' se habrá añadido la conexión al árbol de servidores y se podrá ver la tabla creada anteriormente:



En este punto ya se podrán realizar queries sobre esta tabla con la siguiente opción:



Si se realiza la siguiente consulta se podrán ver todos los datos guardados en la base de datos:

SQLQuery1.sql * - X

CX5130_DB

```
1 SELECT * FROM CX5130
```

100 %

T-SQL Resultados Mensaje

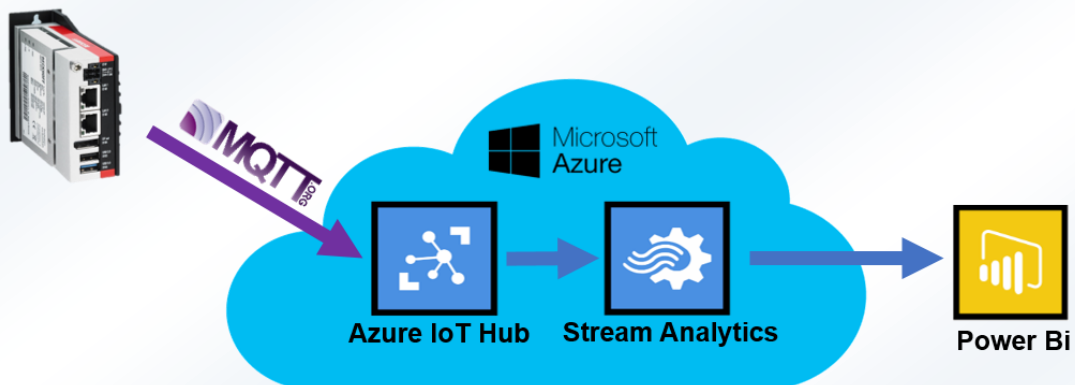
| | Fecha | Temperatura | Humedad |
|----|-------------------------|------------------|--------------------|
| 1 | 2017-10-25 12:38:01.093 | 22.9053173065186 | 4.79301118850708 |
| 2 | 2017-10-25 12:38:06.190 | 76.8108673095703 | 5.98062086105347 |
| 3 | 2017-10-25 12:38:11.290 | 43.486686706543 | 6.65156030654907 |
| 4 | 2017-10-25 12:38:16.390 | 14.8058290481567 | 0.0460822470486164 |
| 5 | 2017-10-25 12:38:21.487 | 84.493782043457 | 5.04493761062622 |
| 6 | 2017-10-25 12:38:26.587 | 27.1335926055908 | 7.82970952987671 |
| 7 | 2017-10-25 12:38:31.687 | 68.7525787353516 | 7.02876329421997 |
| 8 | 2017-10-25 12:38:36.783 | 96.0067138671875 | 3.24193167686462 |
| 9 | 2017-10-25 12:38:41.883 | 4.71198606491089 | 5.28297853469849 |
| 10 | 2017-10-25 12:38:46.980 | 19.6520938873291 | 8.97566223144531 |
| 11 | 2017-10-25 12:38:52.080 | 58.0971984863281 | 9.54009342193604 |

6. VISUALIZACIÓN EN POWER BI

La herramienta de Power Bi de Microsoft permite la visualización de datos de diferentes tipos de fuentes como pueden ser bases de datos, ficheros Excel, etc...

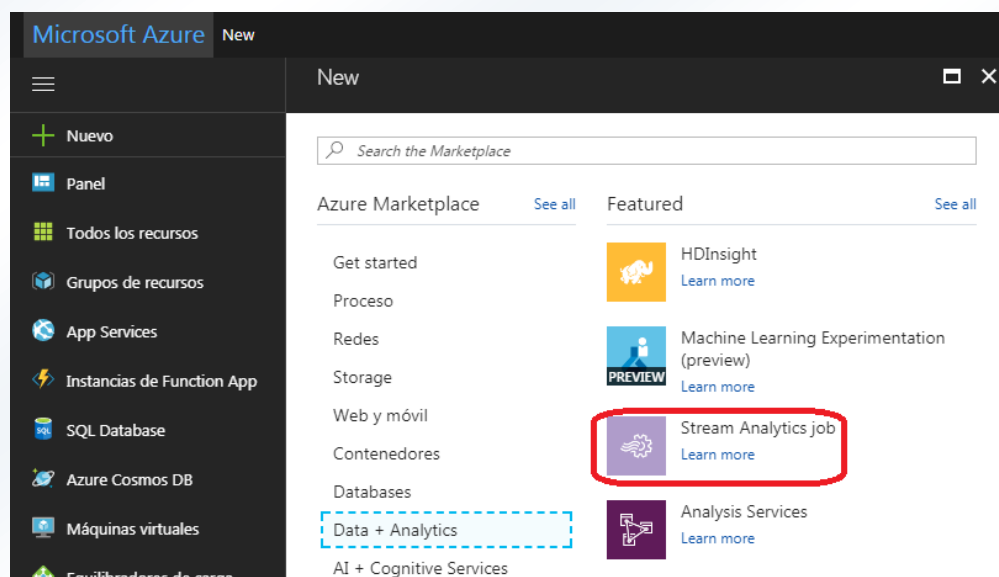
Para visualizar los datos que lleguen a un centro de IoT Hub de Microsoft Azure en Power Bi es necesario hacer uso de un Stream Analytics Job.

La conexión seguiría el siguiente esquema, desde TwinCAT se subirían los datos a un dispositivo de un centro de Azure IoT Hub y una rutina de Stream Analytics leería estos datos y los llevaría a un servicio de Power Bi para poder visualizarlos.



CONFIGURAR EL STREAM ANALYTICS JOB

El primer paso es crear la rutina del Stream Analytics Job que se encargará de leer los datos recibidos del IoT Hub y los volcará a la salida del panel de Power Bi. Para crear este objeto hay que seleccionar la siguiente opción desde el portal de Azure.



En la configuración de este objeto se le dará un nombre y se seleccionará el grupo de recursos a utilizar (en este caso se hospedará en el mismo recurso que el centro de IoT Hub).

Nuevo trabajo de Stream Anal... X

* Nombre de trabajo

IoTHubToPowerBi ✓

* Suscripción

Visual Studio Professional

* Grupo de recursos

☐ Crear nuevo
 ☒ Usar existente

ResourceBeckhoff

* Ubicación

Oeste de Europa

Entorno de hospedaje

☒ Nube
 ☐ Perimetral

☒ Anclar al panel

Crear

Opciones de automatización

Al pulsar sobre el botón de 'Crear' habrá que esperar a la notificación de que se ha creado correctamente:

Notificaciones

Descartar: Informativo Completada Todo

✓ Implementación correcta

20:33

La implementación "Microsoft.StreamAnalyticsJob" se realizó correctamente en el grupo de recursos "ResourceBeckhoff".

Ir al recurso

Anclar al panel

Una vez se ha creado se puede pinchar sobre él desde el panel de control para ir al panel de configuración del Stream Analytics Job.

IoTHubToPowerBi

Trabajo de Stream Analytics

Buscar (Ctrl+I)

Información general

Registro de actividad

Control de acceso (IAM)

Etiquetas

Diagnosticar y solucionar pr...

CONFIGURACIÓN

Bloqueos

TOPOLOGÍA DE TRABAJO

Entradas

Funciones

Consulta

Salidas

Iniciar

Detener

Eliminar

Creado

Información esencial

Grupo de recursos (cambiar)
 ResourceBeckhoff

Estado
 Creado

Ubicación
 West Europe

Nombre de la suscripción (cambiar)
 Visual Studio Professional

Id. de suscripción
 3cad47f2-e1f2-4900-a64c-83d6eb2b7b71

Enviar comentarios
 UserVoice

Creado
 viernes, 29 de diciembre de 2017 20:32:51

Iniciado
 -

Última salida
 -

Entorno de hospedaje
 Nube

Topología de trabajo

Entradas

0

No hay ningún resultado.

Consulta

<>

Salidas

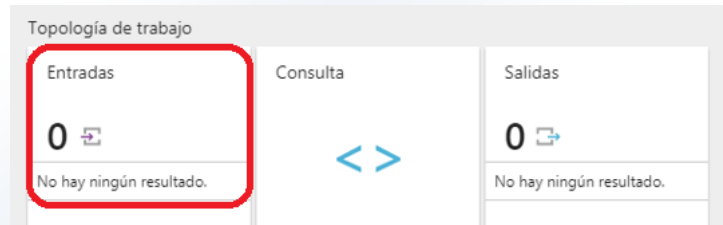
0

No hay ningún resultado.

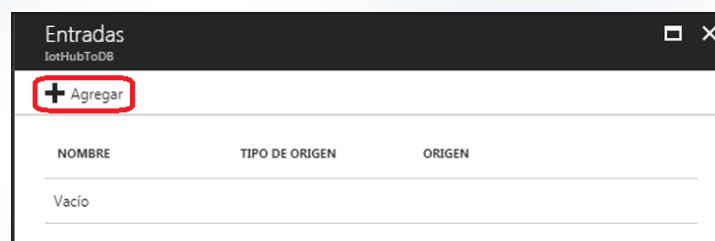
Página 51 de 66

En este panel se puede arrancar y detener el trabajo y se pueden definir los datos de entrada y salida del trabajo y la función que realizará.

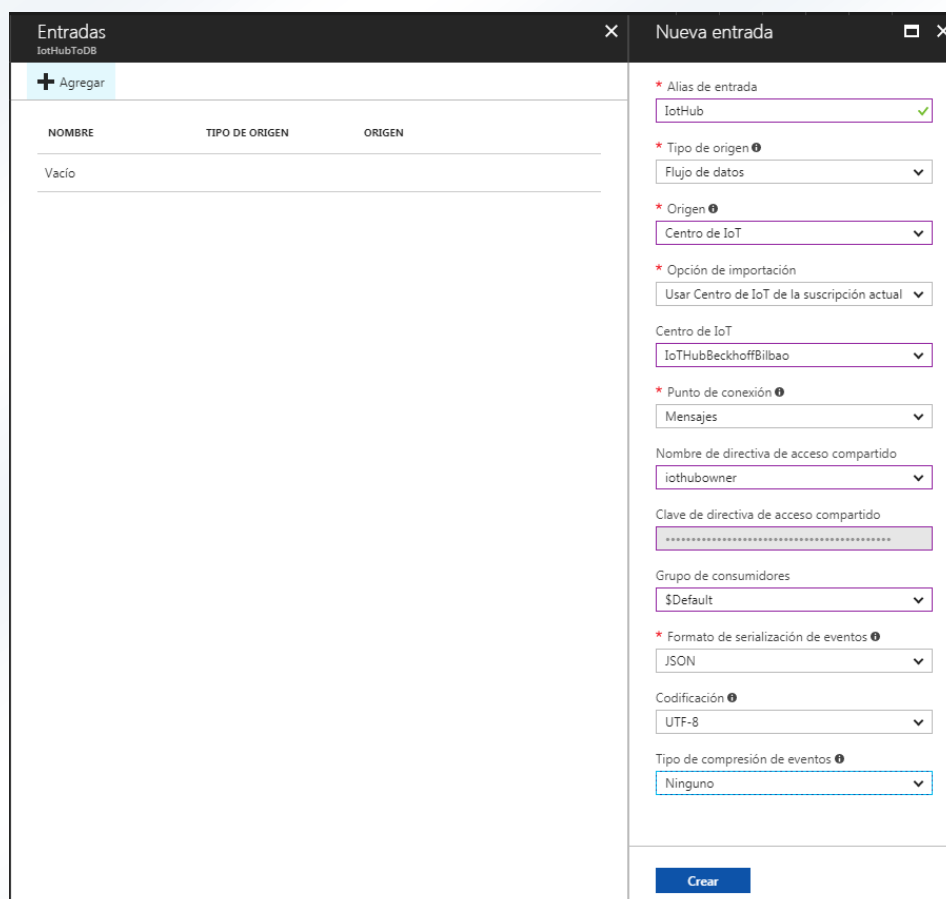
En primer lugar, hay que pulsar sobre el campo de 'Entradas' dentro de la topología para definir la fuente de entrada de los datos:



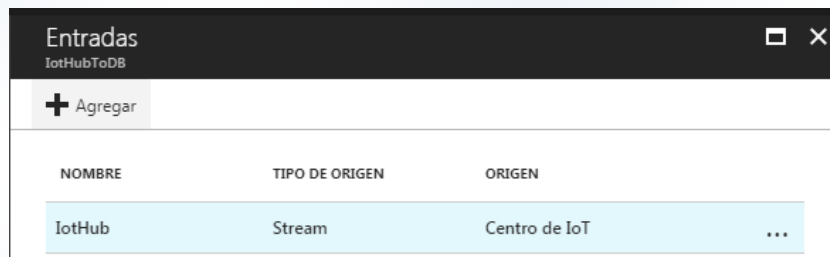
Dentro de la configuración de las entradas hay que pulsar sobre la opción de 'Agregar' para añadir una entrada.



En la configuración de la entrada hay que seleccionar el origen de los datos, en este caso hay que seleccionar el 'Centro de IoT'. En caso de que solo haya un centro de IoT creado, el resto de los datos se rellenarán automáticamente, sino habría que seleccionar el centro del que se quieren recibir los datos para llevarlos a Power Bi.



Al darle a la opción de crear se mostrará este origen de datos en la lista de entradas del Stream Analytics Job.



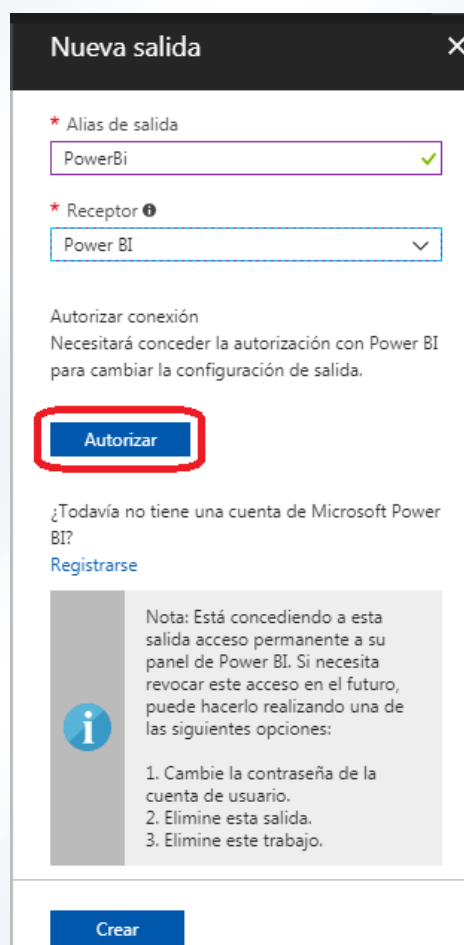
| NOMBRE | TIPO DE ORIGEN | ORIGEN | |
|--------|----------------|---------------|-----|
| IotHub | Stream | Centro de IoT | ... |

El siguiente paso será definir la salida del Stream Analytics Job, para ello hay que pulsar sobre las salidas en la topología del trabajo. Una vez en el listado de las salidas hay que darle a la opción de 'Agregar' para añadir una nueva salida.



| NOMBRE | RECEPTOR |
|--------|----------|
| Vacío | |

En la configuración de las salidas hay que seleccionar como salida el servicio de Power Bi. Al seleccionar el servicio de Power Bi habrá que pulsar sobre el botón de 'Autorizar'.



Nueva salida

* Alias de salida
PowerBi ✓

* Receptor ⓘ
Power BI ▼

Autorizar conexión
Necesitará conceder la autorización con Power BI para cambiar la configuración de salida.

Autorizar

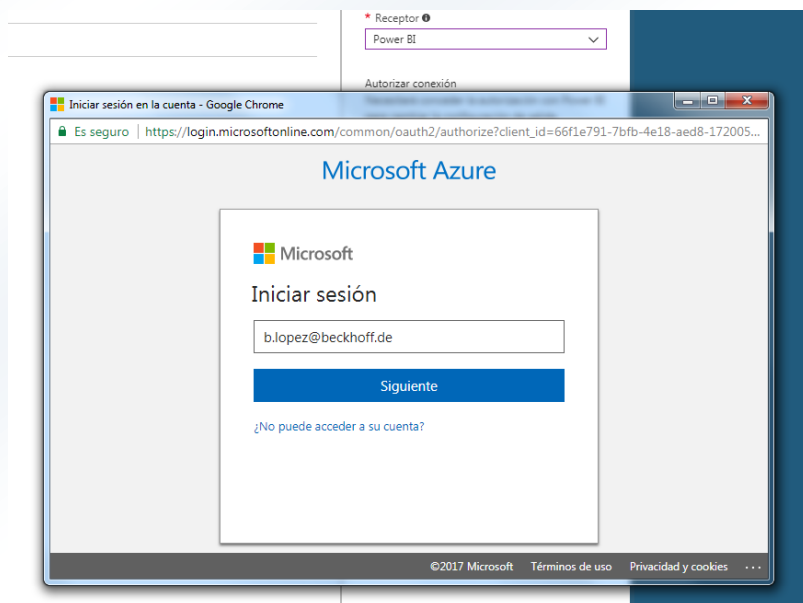
¿Todavía no tiene una cuenta de Microsoft Power BI?
[Registrarse](#)

Nota: Está concediendo a esta salida acceso permanente a su panel de Power BI. Si necesita revocar este acceso en el futuro, puede hacerlo realizando una de las siguientes opciones:

1. Cambie la contraseña de la cuenta de usuario.
2. Elimine esta salida.
3. Elimine este trabajo.

Crear

Al pulsar sobre el botón de autorizar habrá que loguearse con una cuenta en la que estemos registrados en Power Bi (se puede utilizar una cuenta gratuita para lo que se va a hacer en este ejemplo). Al pulsar el botón de 'Autorizar' aparecerá la siguiente ventana del navegador para realizar el login.



Una vez logueados habrá que definir el nombre del conjunto de datos y de la tabla que se creará en Power Bi con los datos del dispositivo IoT Hub. Si el conjunto de datos existiese se sobrescribiría al ejecutarse esta función, por lo que, lo más sencillo es no crear el conjunto en Power Bi y que se genere automáticamente a través de este Stream Analytics Job.

Nueva salida

* Alias de salida

PowerBi

✓

* Receptor

Power BI

▼

Área de trabajo de grupo

Mi área de trabajo

▼

* Nombre del conjunto de datos

DatosCX

✓

⚠

En el caso de que el conjunto de datos o la tabla ya existan en la suscripción de Microsoft Power BI, se sobrescribirán.

* Nombre de la tabla

Temperatura_Humedad

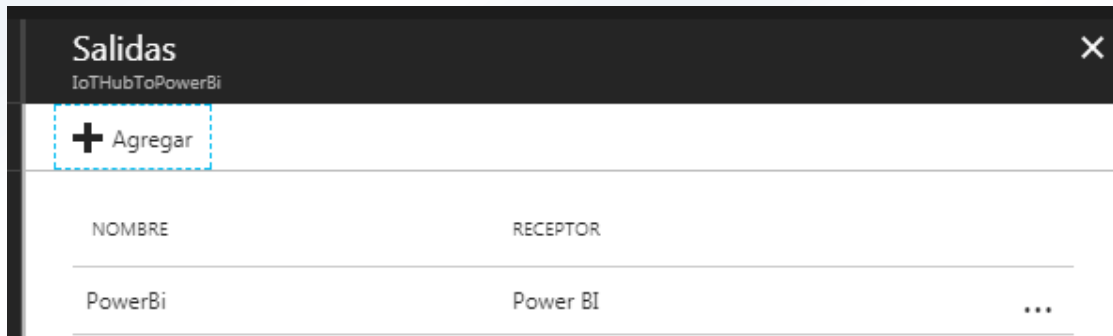
✓

Autorizado actualmente como Borja Lopez

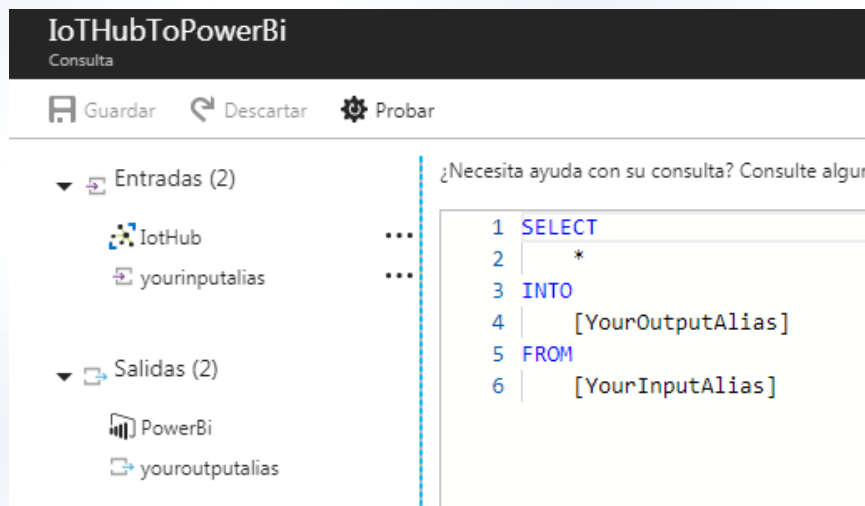
(b.lopez@beckhoff.de)

Crear

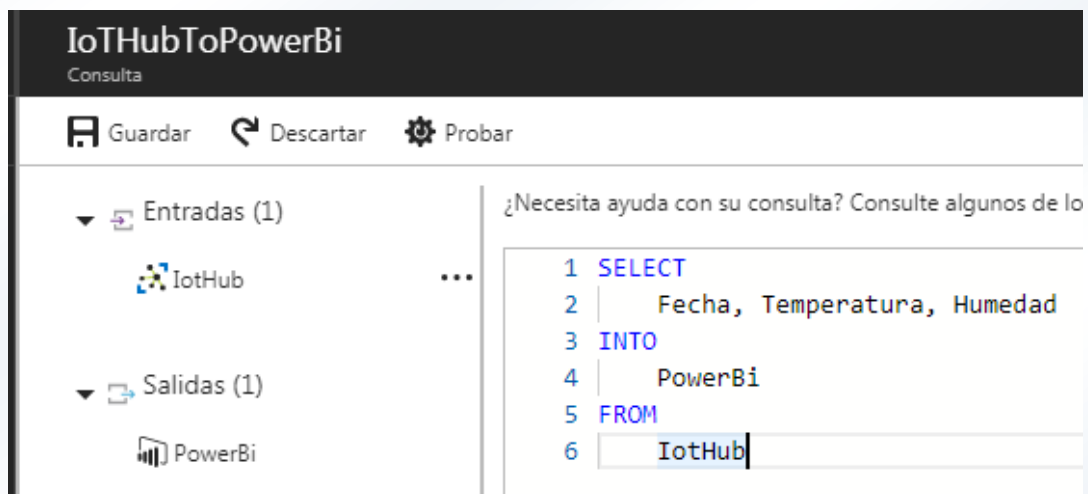
Al darle al botón de 'Crear' se mostrará esta salida de datos en la lista de salidas del Stream Analytics Job.



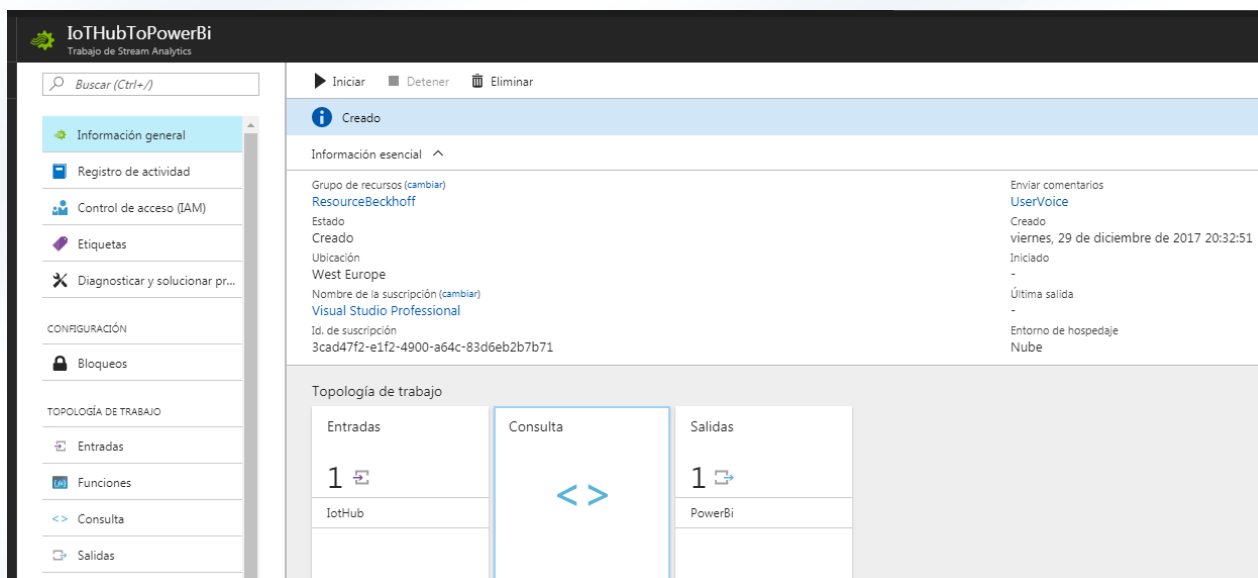
El último paso será definir la consulta SQL del trabajo. Al pinchar sobre las consultas para crearla aparecerá el siguiente código base:



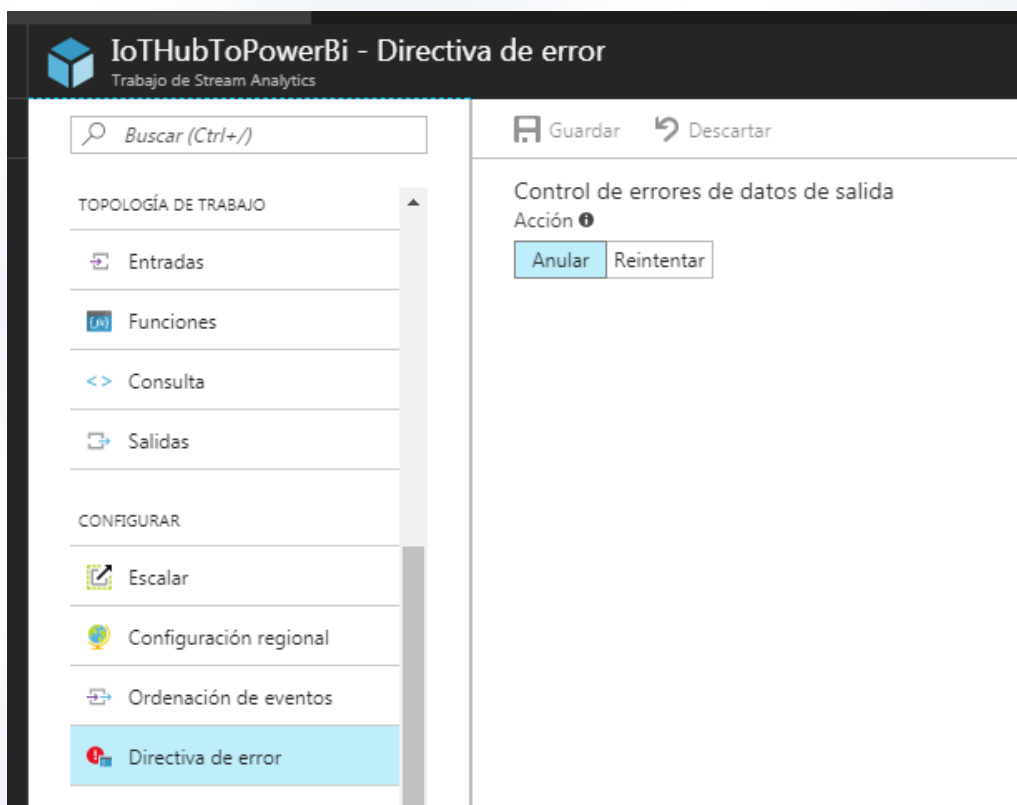
El código que hay que añadir es el de la siguiente imagen. En él se cogen los campos 'Fecha', 'Temperatura' y 'Humedad' del mensaje JSON que se recibirá en el centro de IoT y se escribirán sobre la salida de Power Bi.



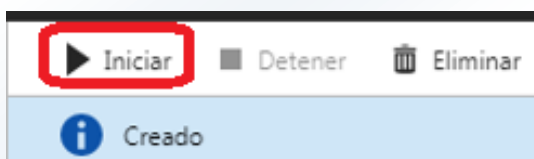
Una vez guardada la consulta se puede ver en el panel de administración del Stream Analytics Job todos los parámetros configurados.



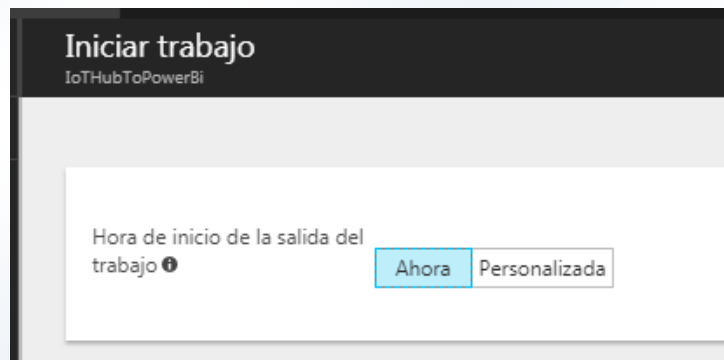
Hay que tener en cuenta que si en el telegrama no se recibiese alguno de los campos definidos en la consulta dentro del JSON (por ejemplo, si no se recibe el dato de 'Temperatura') el trabajo entraría en error y dejaría de ejecutarse. Para evitar esto se puede seleccionar la opción de 'Anular' dentro del menú de 'Directiva de error' y darle a 'Guardar':



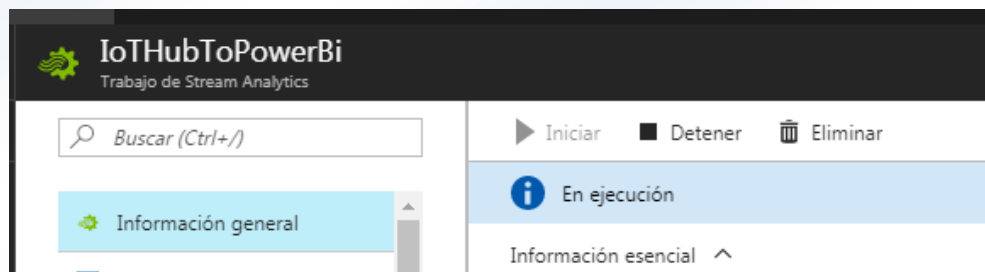
Una vez esté todo correctamente configurado hay que darle a la opción de 'Iniciar':



Si fuese necesario se podría aplazar la hora para iniciar el trabajo, en este caso se seleccionará la opción 'Ahora' para que se inicie inmediatamente:



Una vez se haya iniciado el trabajo se puede comprobar su estado dentro de la 'Información general' del Stream Analytics Job.

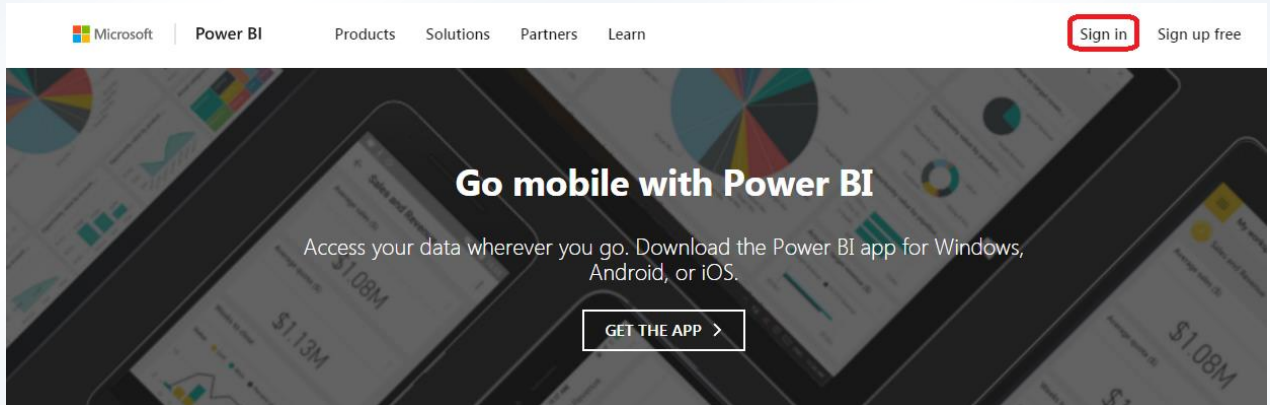


En esta misma sección también se puede comprobar el número de eventos de entrada (mensajes recibidos al IoT Hub), eventos de salida (escritura en el servicio de Power Bi) y errores que se hayan producido.

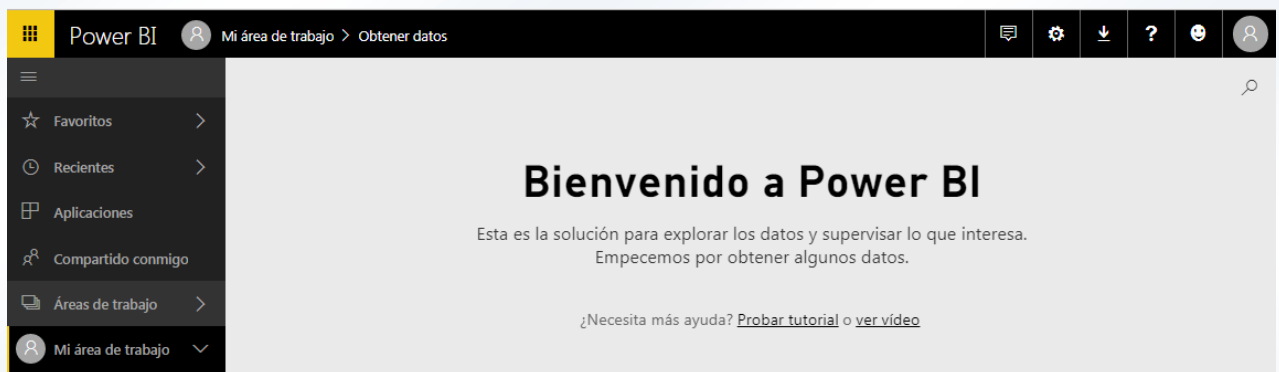


CONFIGURACIÓN DEL PANEL DE CONTROL EN POWER BI

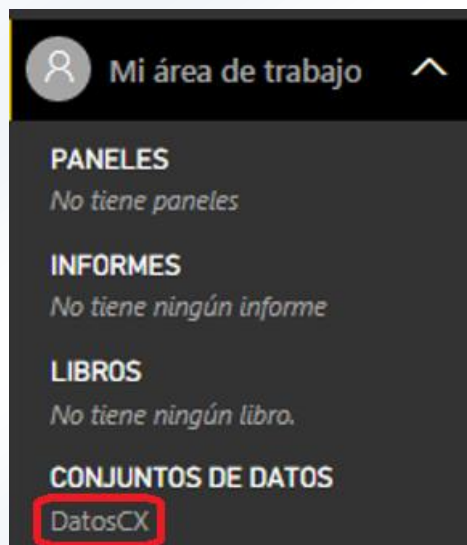
La configuración del panel de control de Power Bi se hace mediante una herramienta web en el siguiente [enlace](#). Una vez se esté en la página web hay que loguearse con la misma cuenta que se ha configurado en la salida de Power Bi del Stream Analytics Job



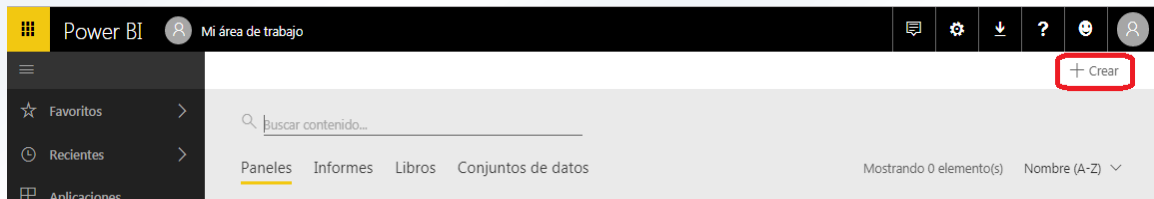
Una vez se esté logueado se verá el siguiente panel:



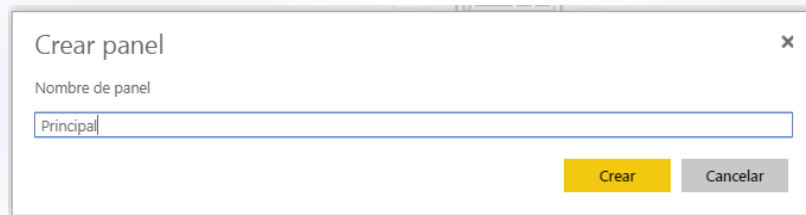
Si ya se ha iniciado la subida de datos y el Stream Analytics Job ha escrito algún dato sobre la salida de Power Bi, en la sección de 'Mi área de trabajo' se debería ver el conjunto de datos definido como salida del Stream Analytics Job:



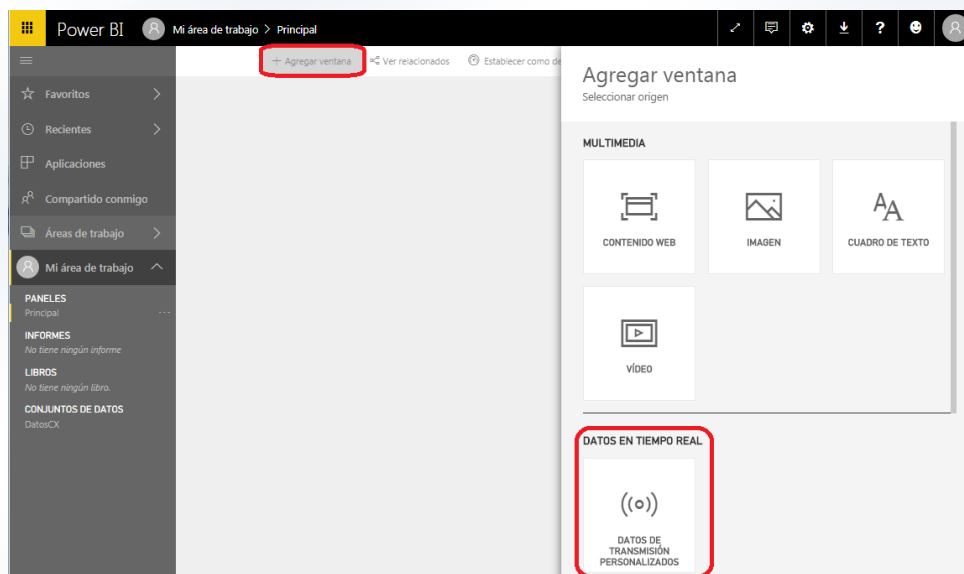
Una vez se tienen los datos se va a crear un panel de control donde se puedan ver los datos en tiempo real. Por ello, el primer paso es crear un nuevo panel sobre 'Mi área de trabajo':



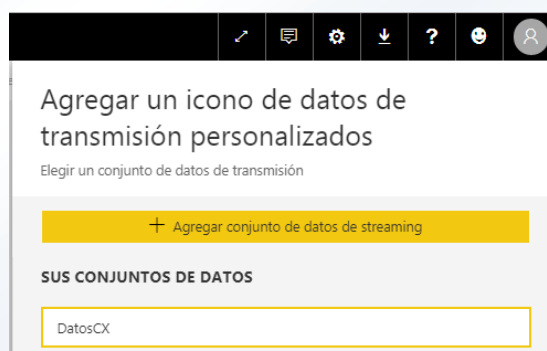
Se le debe dar un nombre a este panel, en este caso 'Principal', y pulsar sobre 'Crear':



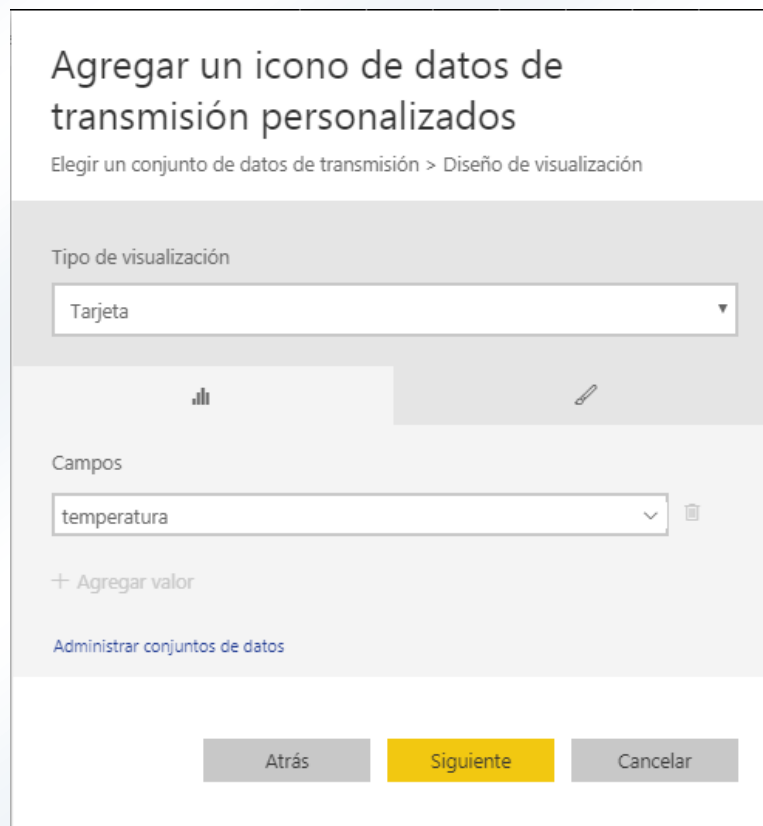
En Power Bi se pueden realizar diferentes tipos de visualizaciones de los datos, por ejemplo, visualizaciones en tiempo real con el 'Power Bi Streaming' o visualización de datos con un refresco más lento a través de los informes. En primer lugar, se va a crear una fuente de datos en tiempo real para la visualización de la variable 'Temperatura'. Para ello, sobre el panel recién creado, hay que pulsar sobre el botón de '+ Agregar ventana'.



En el asistente que aparecerá se podrá seleccionar el conjunto de datos que se recibe del Stream Analytics Job:



En primer lugar, únicamente se quiere ver el valor actual como un numérico, por lo que se elige el tipo de visualización 'Tarjeta' y se selecciona el campo 'temperatura' para que se muestre su valor:



Agregar un icono de datos de transmisión personalizados

Elegir un conjunto de datos de transmisión > Diseño de visualización

Tipo de visualización

Tarjeta

Campos

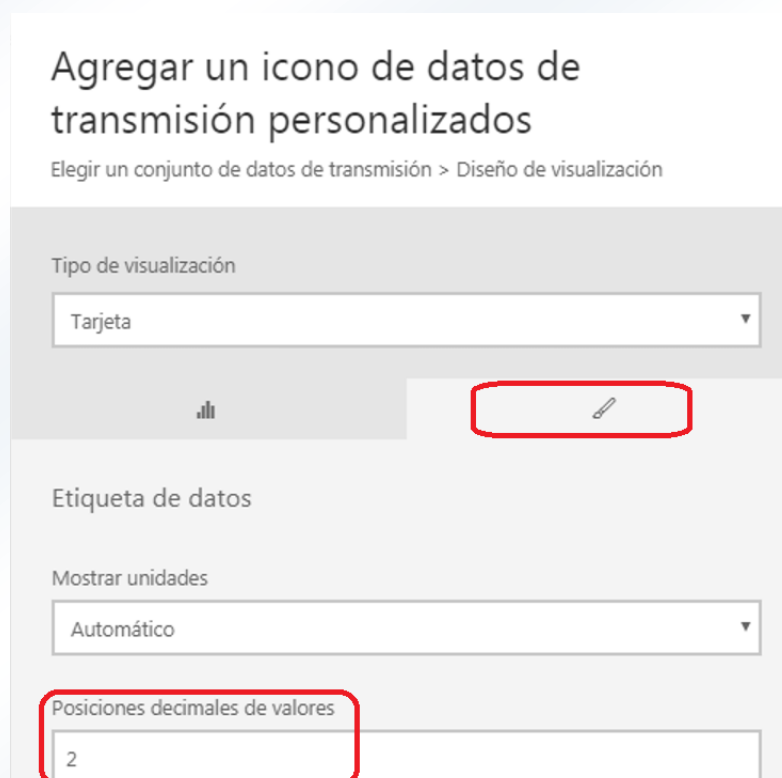
temperatura

+ Agregar valor

[Administrar conjuntos de datos](#)

Atrás Siguiente Cancelar

Si se quiere modificar el número de decimales que se van a mostrar o las unidades se puede hacer desde el siguiente panel:



Agregar un icono de datos de transmisión personalizados

Elegir un conjunto de datos de transmisión > Diseño de visualización

Tipo de visualización

Tarjeta

Etiqueta de datos

Mostrar unidades

Automático

Posiciones decimales de valores

2

Una vez configurado se podrá cambiar el título que se va a mostrar en la tarjeta y al darle a 'Aplicar' se verá una tarjeta con el valor actual de la temperatura.



Del mismo modo se podría añadir un gráfico que muestre los últimos valores de la temperatura. Para ello habría que seguir los mismos pasos, pero en el tipo de visualización hay que elegir 'Gráfico de líneas' y luego poner las variables para el eje X (la fecha) y el eje Y (la temperatura). También hay que seleccionar la antigüedad de los datos que se quiere mostrar:

Agregar un icono de datos de transmisión personalizados

Elegir un conjunto de datos de transmisión > Diseño de visualización

Tipo de visualización
Gráfico de líneas

Eje
fecha

+ Agregar valor

Leyenda
+ Agregar valor

Valores
temperatura

+ Agregar valor

Periodo de tiempo para mostrar
Último periodo de tiempo 10 Minutos

[Administrar conjuntos de datos](#)

Atrás **Siguiente** Cancelar

Al darle a siguiente se puede especificar el título y el subtítulo que se va a mostrar sobre el grafico.

Detalles del icono

* Requerido

Detalles

☒ Mostrar el título y el subtítulo

Título
Grafico de temperatura

Subtítulo
Muestreado cada 3 segundos

Funcionalidad
☐ Establecer vínculo personalizado

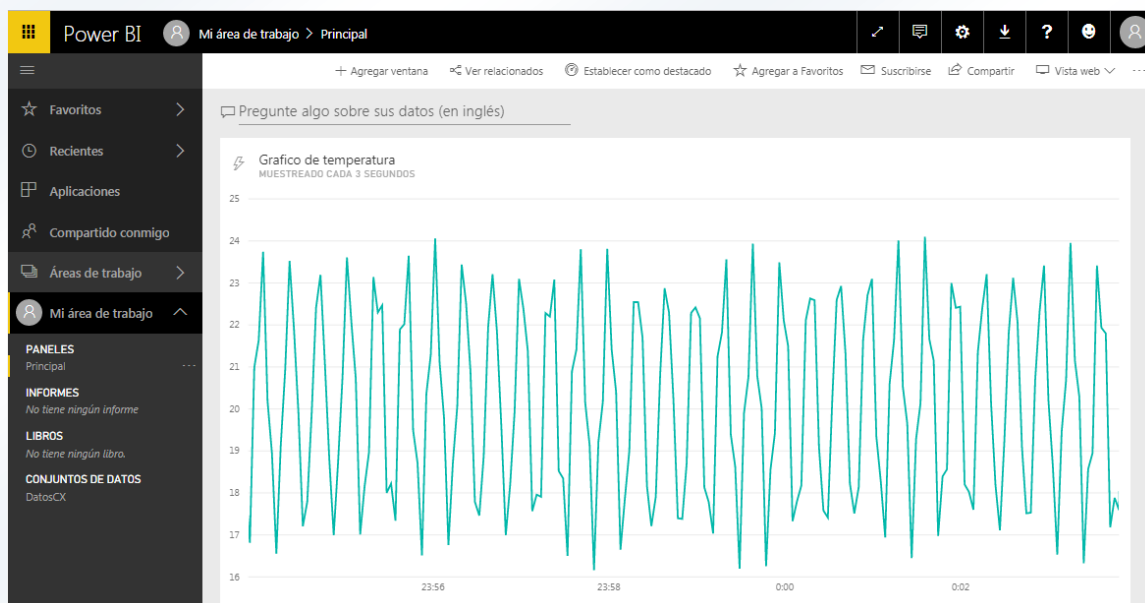
Tipo de vínculo
☐ Vínculo externo
☒ Vínculo a un panel o informe en el área de trabajo actual

Panel o informe al que vincular: *

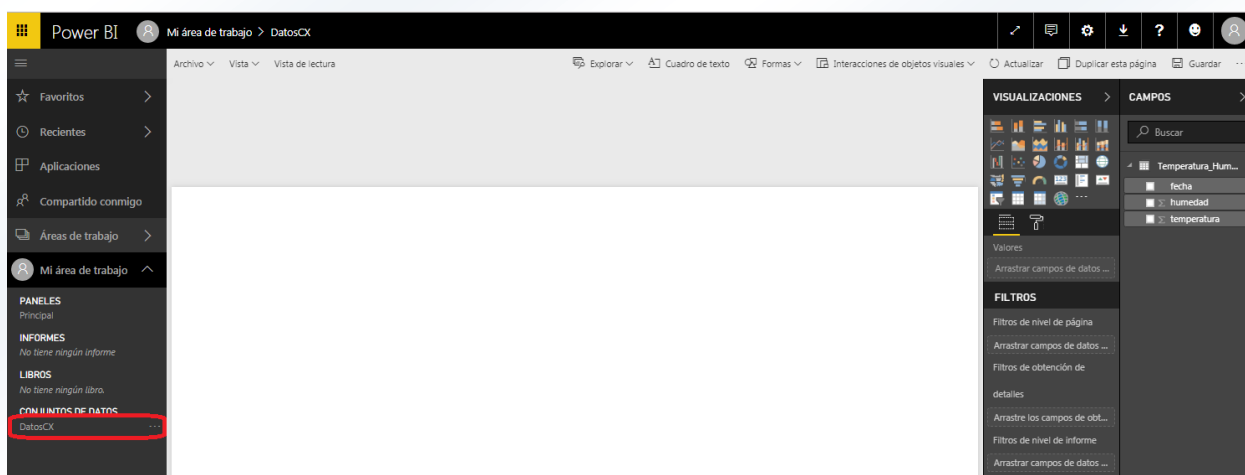
[Restaurar valores predeterminados](#)
[Detalles técnicos](#)

Atrás **Aplicar** Cancelar

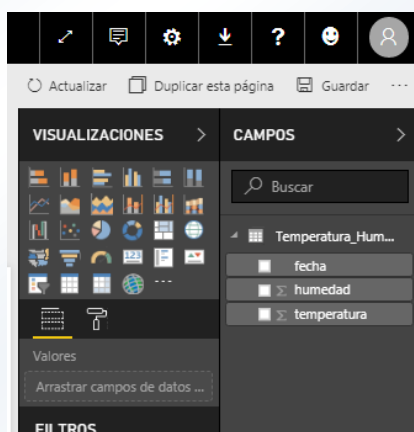
Una vez configurado, al darle a ‘Aplicar’ aparecerá el grafico con los datos recibidos:



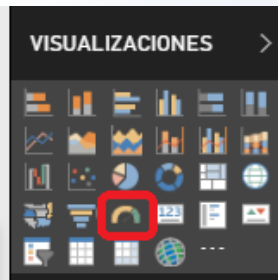
En algún caso puede interesar añadir al panel de control otros elementos relativos a históricos, como por ejemplo la media, los valores máximos y mínimos o una gráfica con todos los datos recibidos. Para hacer esto es necesario añadir un informe sobre el conjunto de datos. Para ello hay que pinchar sobre el conjunto de datos y se abrirá la siguiente pantalla:



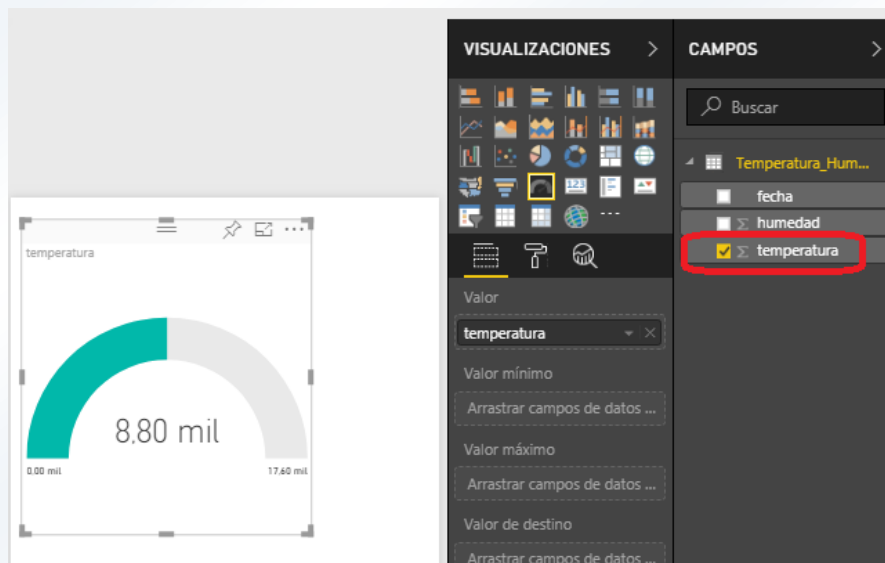
A la derecha del panel se pueden ver los datos disponibles en el conjunto (las variables ‘fecha’, ‘temperatura’ y ‘humedad’) y los objetos que se pueden añadir:



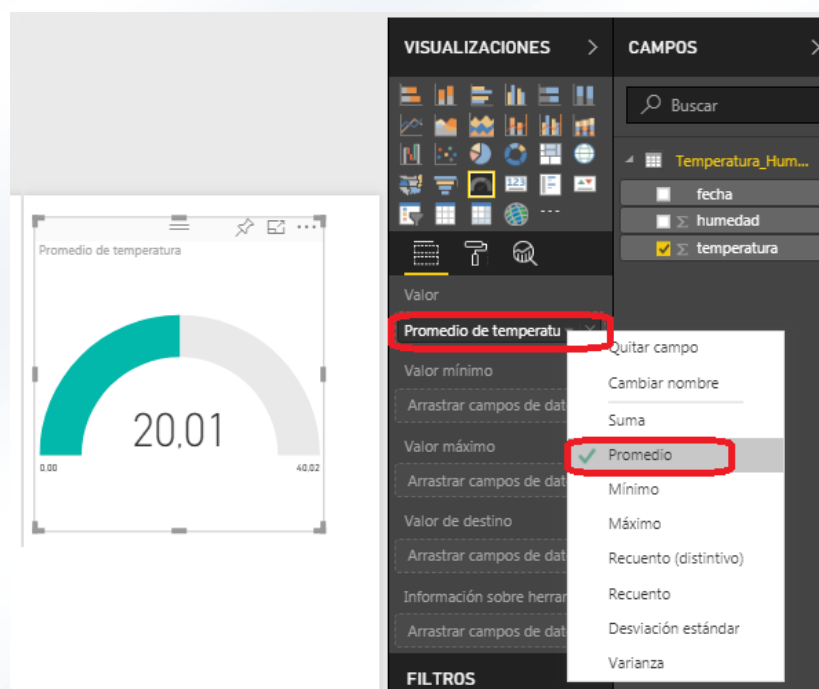
En primer lugar, se va a añadir un medidor que muestre la media de todos los valores medidos de la temperatura. Para ello hay que pinchar sobre el siguiente objeto:



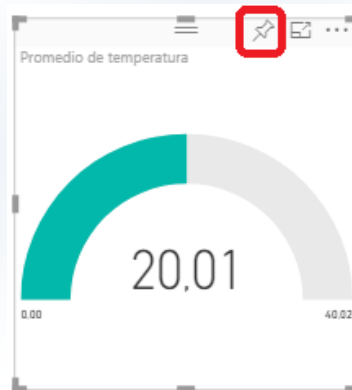
Una vez añadido el objeto y seleccionado en el informe hay que seleccionar el campo que se quiere que muestre, en este caso la variable 'temperatura':



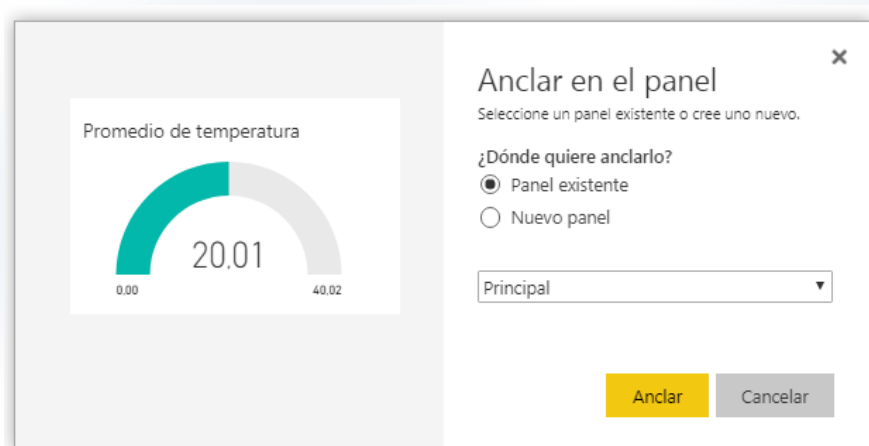
Por defecto se mostrará la suma de todos los registros de la temperatura, pero se puede cambiar el tipo de operación que se quiere que se muestre:



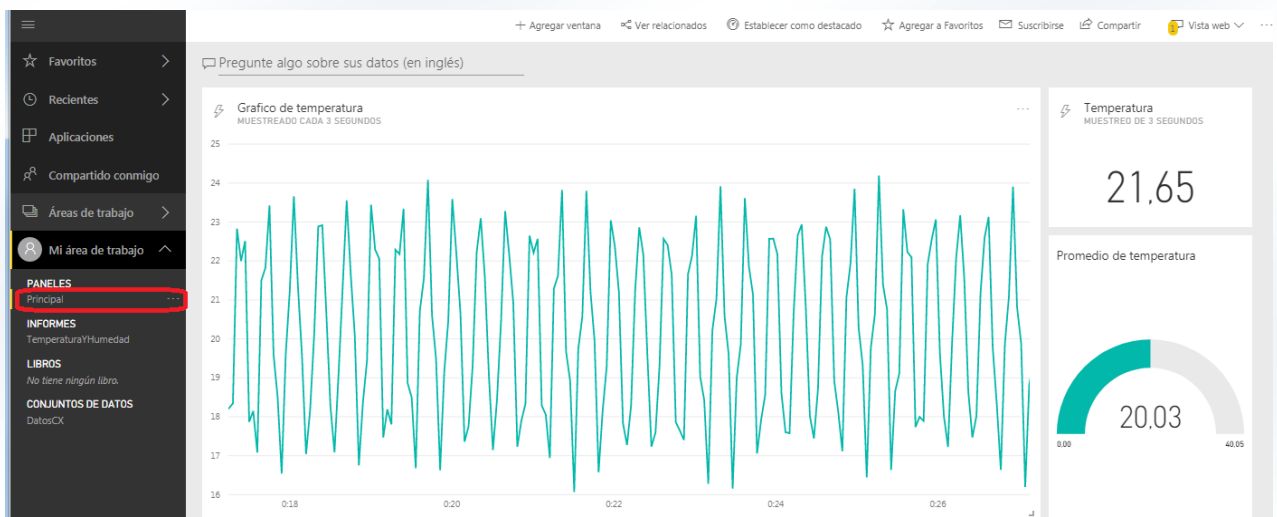
También se pueden modificar diferentes opciones del objeto como colores, títulos, formato de los valores, etc... Una vez esté configurado al gusto se puede anclar al panel creado anteriormente (el panel 'Principal' que muestra los valores en tiempo real de la variable 'Temperatura'). Para ello hay que anclar el objeto con el siguiente botón:



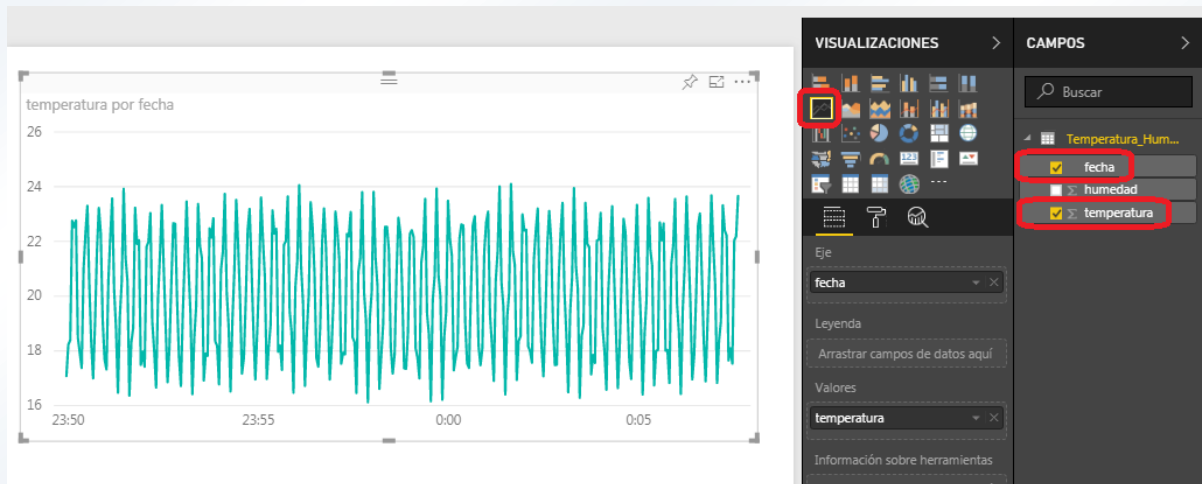
Al pulsar sobre el botón de anclar pedirá guardar un informe con este objeto. A este informe se le podrán añadir más elementos de visualización y se podrán compartir a través de un link con más gente, pero el refresco que tendrá no será tan rápido como en un panel. Una vez guardado aparecerá una ventana para seleccionar sobre que panel se va a añadir este objeto, en este caso hay que elegir el panel ya creado con el nombre de 'Principal':



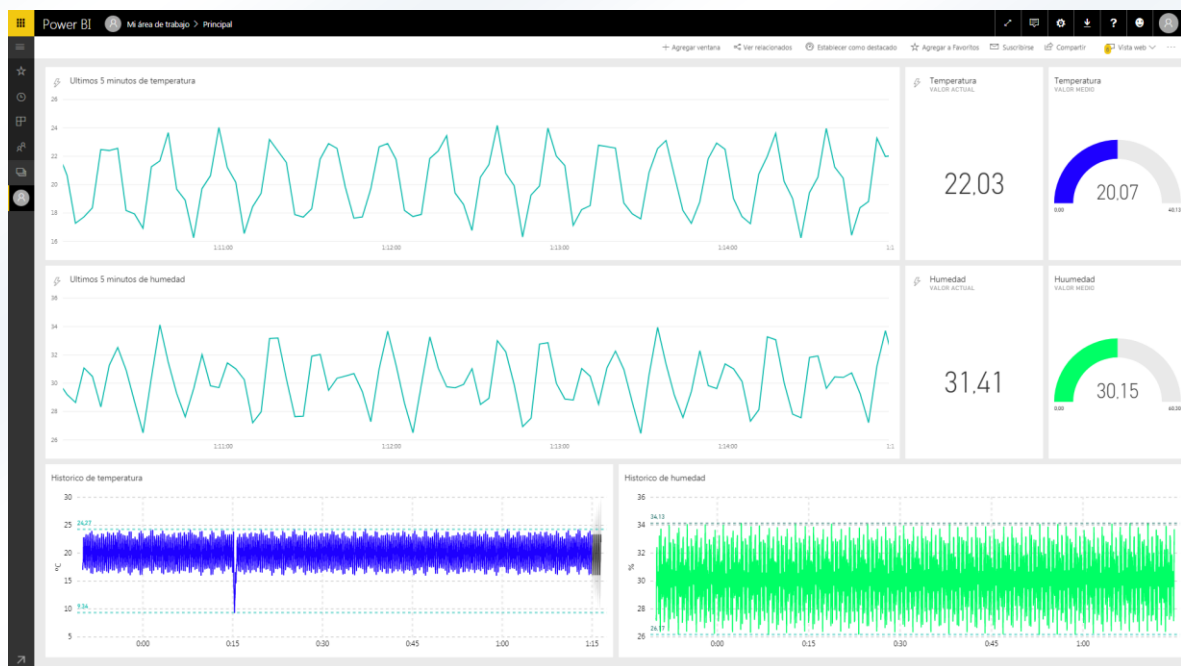
Una vez anclado, si se va al panel 'Principal' se verá este objeto añadido:



Del mismo modo se podría añadir una gráfica con los históricos. Para ello hay que seleccionar el siguiente objeto y poner las variables para el eje X y el eje Y sobre el informe y, finalmente, anclarlo al panel.



Juntando todos estos tipos de objetos se puede configurar un panel de control más completo. Este panel se podría compartir con más gente en caso de tener una cuenta de Power Bi Pro, así como hacer informes que se actualicen automáticamente cada cierto tiempo.



Por último, Power Bi tiene aplicaciones para plataformas móviles de forma que se podría ver este mismo panel en el teléfono móvil o tablet. La descarga de la aplicación para el móvil se puede hacer desde el siguiente [enlace](#).

Microsoft
Power BI
Productos
Soluciones
Asociados
Más información
Iniciar sesión
Registrarse de forma gratuita

Asegúrese de que los datos lo acompañan en sus viajes

Permanezca conectado a los datos desde cualquier lugar y en cualquier momento con la aplicación Power BI para Windows, iOS y Android. Tenga al alcance de la mano una visión global de sus datos empresariales en cualquier lugar.

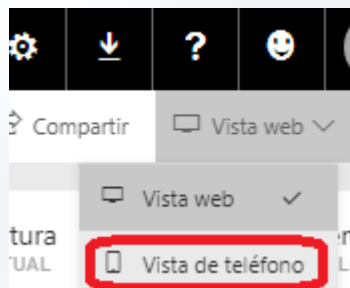
Descargar de la Tienda Windows

Descargar de la App store

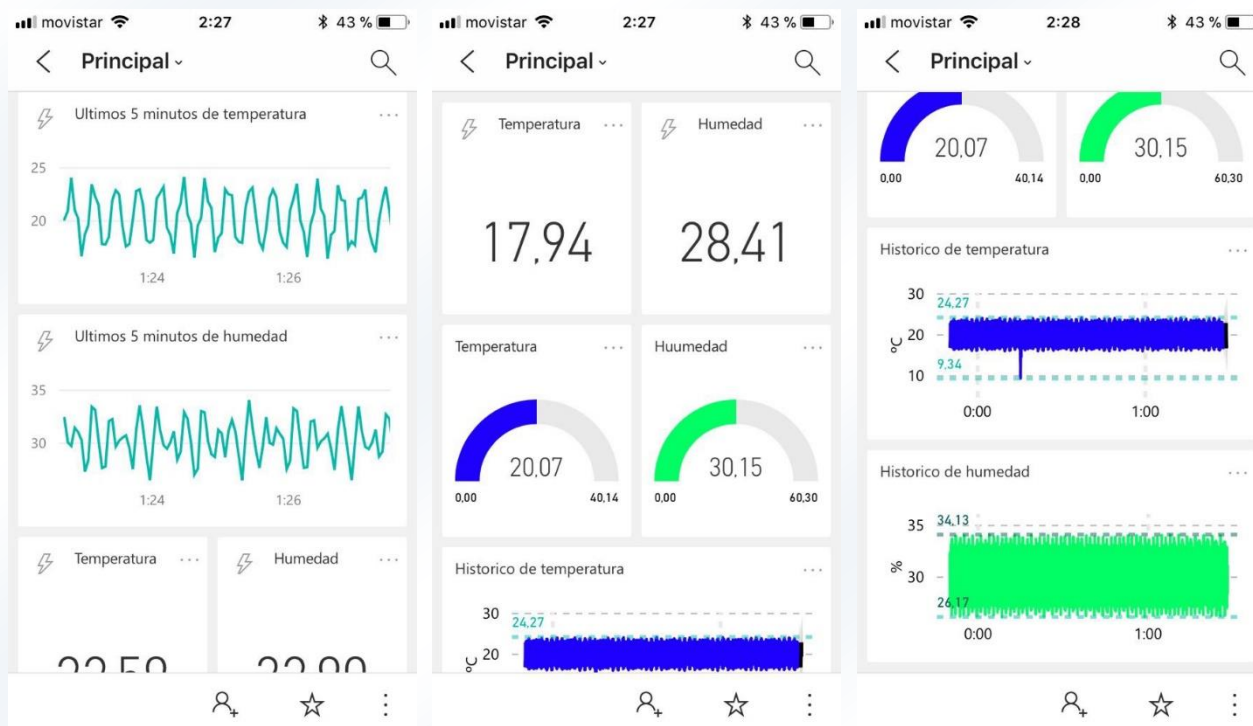
Descargar de la Google Play

Disponible también en otros países ►

En el mismo panel se puede modificar el orden y la posición que tendrá cada objeto visto desde un móvil con la siguiente opción:



El resultado sería algo similar a las siguientes imágenes:



ANNEX 9

TAULA DE SENYALS DEL SCADA

| N° | Origin | Data Structure | Service | Variable Name | TAG | Type | FROM Modbus Address (DEC) | TO Modbus Address (DEC) | FROM Modbus Address (HEX) | TO Modbus Address (HEX) | Max | Min | Default | Unit |
|----|--------|----------------------------|----------------------------|---------------------|-----------------------------------|------|------------------------------|----------------------------|------------------------------|----------------------------|----------|-----------|---------|---------|
| 1 | FLC | EtherCAT_Commands_Modbus | CommonBlock | | FLCcmdMB_COMM | UINT | 12288 | | 3000 | | 65535 | 0 | 0 | CMD |
| 2 | FLC | EtherCAT_Commands_Modbus | InertiaEmulation | | FLCcmdMB_IE | UINT | 12289 | | 3001 | | 65535 | 0 | 0 | CMD |
| 3 | FLC | EtherCAT_Commands_Modbus | FrequencyRegulation | | FLCcmdMB_FR | UINT | 12290 | | 3002 | | 65535 | 0 | 0 | CMD |
| 4 | FLC | EtherCAT_Commands_Modbus | ModalFreq | | FLCcmdMB_MFA | UINT | 12291 | | 3003 | | 65535 | 0 | 0 | CMD |
| 5 | FLC | EtherCAT_Commands_Modbus | VoltageRegulation | | FLCcmdMB_VR | UINT | 12292 | | 3004 | | 65535 | 0 | 0 | CMD |
| 6 | FLC | EtherCAT_Commands_Modbus | AdditionalService | | FLCcmdMB_AddServ | UINT | 12293 | | 3005 | | 65535 | 0 | 0 | CMD |
| 7 | FLC | EtherCAT_Commands_EAP | CommonBlock_EAP | | FLCcmdEAP_COMM | UINT | 12294 | | 3006 | | 65535 | 0 | 0 | CMD |
| 8 | FLC | EtherCAT_Commands_EAP | InertiaEmulation_EAP | | FLCcmdEAP_IE | UINT | 12295 | | 3007 | | 65535 | 0 | 0 | CMD |
| 9 | FLC | EtherCAT_Commands_EAP | FrequencyRegulation_EAP | | FLCcmdEAP_FR | UINT | 12296 | | 3008 | | 65535 | 0 | 0 | CMD |
| 10 | FLC | EtherCAT_Commands_EAP | ModalFreq_EAP | | FLCcmdEAP_MFA | UINT | 12297 | | 3009 | | 65535 | 0 | 0 | CMD |
| 11 | FLC | EtherCAT_Commands_EAP | VoltageRegulation_EAP | | FLCcmdEAP_VR | UINT | 12298 | | 300A | | 65535 | 0 | 0 | CMD |
| 12 | FLC | EtherCAT_Commands_EAP | Adittional_EAP | | FLCcmdEAP_AddServ | UINT | 12299 | | 300B | | 65535 | 0 | 0 | CMD |
| 13 | SLC | SecondLevel_Commands | CommonComands | | SLCcmd_COMM | UINT | 12300 | | 300C | | 65535 | 0 | 0 | CMD |
| 14 | SLC | SecondLevel_Commands | ResourceVariability | | SLCcmd_RVM | UINT | 12301 | | 300D | | 65535 | 0 | 0 | CMD |
| 15 | SLC | SecondLevel_Commands | PowerShiftingManage | | SLCcmd_PSM | UINT | 12302 | | 300E | | 65535 | 0 | 0 | CMD |
| 16 | SLC | SecondLevel_Commands | RampRateEnhancemet | | SLCcmd_RRE | UINT | 12303 | | 300F | | 65535 | 0 | 0 | CMD |
| 17 | SLC | SecondLevel_Commands | PowerCurtailment | | SLCcmd_PCM | UINT | 12304 | | 3010 | | 65535 | 0 | 0 | CMD |
| 18 | SLC | SecondLevel_Commands | PowerBalanceProv | | SLCcmd_PBP | UINT | 12305 | | 3011 | | 65535 | 0 | 0 | CMD |
| 19 | SLC | SecondLevel_Commands | AdditionalService | | SLCcmd_AddServ | UINT | 12306 | | 3012 | | 65535 | 0 | 0 | CMD |
| 20 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | NominalPower | FLCparamMB_COMM_NominalPower | REAL | 12307 | 12308 | 3013 | 3014 | 1.00E+06 | -1.00E+06 | 1000000 | Watts |
| 21 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | NominalVoltage | FLCparamMB_COMM_NominalVoltage | REAL | 12309 | 12310 | 3015 | 3016 | 400 | 0 | 240 | Voltage |
| 22 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | NominalFreq | FLCparamMB_COMM_NominalFreq | REAL | 12311 | 12312 | 3017 | 3018 | 60 | 0 | 50 | Hz |
| 23 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | ActivePowerLimit | FLCparamMB_COMM_ActivePowerLimit | REAL | 12313 | 12314 | 3019 | 301A | 1000000 | 0 | 1000000 | Watts |
| 24 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | ReactivePowerLimit | FLCparamMB_COMM_ReactivePowerLim | REAL | 12315 | 12316 | 301B | 301C | 1000000 | 0 | 1000000 | Watts |
| 25 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved1 | FLCparamMB_COMM_RES01 | REAL | 12317 | 12318 | 301D | 301E | | | | |
| 26 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved2 | FLCparamMB_COMM_RES02 | REAL | 12319 | 12320 | 301F | 3020 | | | | |
| 27 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved3 | FLCparamMB_COMM_RES03 | REAL | 12321 | 12322 | 3021 | 3022 | | | | |
| 28 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved4 | FLCparamMB_COMM_RES04 | REAL | 12323 | 12324 | 3023 | 3024 | | | | |
| 29 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved5 | FLCparamMB_COMM_RES05 | REAL | 12325 | 12326 | 3025 | 3026 | | | | |
| 30 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved6 | FLCparamMB_COMM_RES06 | REAL | 12327 | 12328 | 3027 | 3028 | | | | |
| 31 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved7 | FLCparamMB_COMM_RES07 | REAL | 12329 | 12330 | 3029 | 302A | | | | |
| 32 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved8 | FLCparamMB_COMM_RES08 | REAL | 12331 | 12332 | 302B | 302C | | | | |
| 33 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved9 | FLCparamMB_COMM_RES09 | REAL | 12333 | 12334 | 302D | 302E | | | | |
| 34 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved10 | FLCparamMB_COMM_RES10 | REAL | 12335 | 12336 | 302F | 3030 | | | | |
| 35 | FLC | EtherCAT_Parameters_Modbus | CommonBlock_Params | Reserved11 | FLCparamMB_COMM_RES11 | REAL | 12337 | 12338 | 3031 | 3032 | | | | |
| 36 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Inertia_Constant | FLCparamMB_IE_Inertia_Constant | REAL | 12339 | 12340 | 3033 | 3034 | 20 | 0 | 5 | s |
| 37 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | HighPassFilterConst | FLCparamMB_IE_HighPassFilterConst | REAL | 12341 | 12342 | 3035 | 3036 | 10 | -10 | 0.1 | Hz |
| 38 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Max_ROCOF | FLCparamMB_IE_Max_ROCOF | REAL | 12343 | 12344 | 3037 | 3038 | 10 | -10 | 5 | MW/s |
| 39 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | PowerLimit_Inertia | FLCparamMB_IE_PowerLimit_Inertia | REAL | 12345 | 12346 | 3039 | 303A | 1.00E+06 | 0.00E+00 | 1000000 | Watts |
| 40 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved1 | FLCparamMB_IE_RES01 | REAL | 12347 | 12348 | 303B | 303C | | | | |
| 41 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved2 | FLCparamMB_IE_RES02 | REAL | 12349 | 12350 | 303D | 303E | | | | |
| 42 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved3 | FLCparamMB_IE_RES03 | REAL | 12351 | 12352 | 303F | 3040 | | | | |
| 43 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved4 | FLCparamMB_IE_RES04 | REAL | 12353 | 12354 | 3041 | 3042 | | | | |
| 44 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved5 | FLCparamMB_IE_RES05 | REAL | 12355 | 12356 | 3043 | 3044 | | | | |
| 45 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved6 | FLCparamMB_IE_RES06 | REAL | 12357 | 12358 | 3045 | 3046 | | | | |
| 46 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved7 | FLCparamMB_IE_RES07 | REAL | 12359 | 12360 | 3047 | 3048 | | | | |
| 47 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved8 | FLCparamMB_IE_RES08 | REAL | 12361 | 12362 | 3049 | 304A | | | | |
| 48 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved9 | FLCparamMB_IE_RES09 | REAL | 12363 | 12364 | 304B | 304C | | | | |
| 49 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved10 | FLCparamMB_IE_RES10 | REAL | 12365 | 12366 | 304D | 304E | | | | |
| 50 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved11 | FLCparamMB_IE_RES11 | REAL | 12367 | 12368 | 304F | 3050 | | | | |
| 51 | FLC | EtherCAT_Parameters_Modbus | InertiaEmulation_Params | Reserved12 | FLCparamMB_IE_RES12 | REAL | 12369 | 12370 | 3051 | 3052 | | | | |
| 52 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | DroopConstantP | FLCparamMB_FR_DroopConstantP | REAL | 12371 | 12372 | 3053 | 3054 | 10 | 0 | 4 | % |
| 53 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Max_delta_Energy | FLCparamMB_FR_Max_delta_Energy | REAL | 12373 | 12374 | 3055 | 3056 | 1000000 | -1000000 | 1000000 | Watts |
| 54 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | PowerLimit_Freq | FLCparamMB_FR_PowerLimit_Freq | REAL | 12375 | 12376 | 3057 | 3058 | 1000000 | 0 | 1000000 | Wats |
| 55 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | FrequencyCutOff | FLCparamMB_FR_FrequencyCutOff | REAL | 12377 | 12378 | 3059 | 305A | 10 | 0 | 0.2 | Hz |
| 56 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | FreqDeadBand | FLCparamMB_FR_FreqDeadBand | REAL | 12379 | 12380 | 305B | 305C | 100 | -100 | 30 | mHz |
| 57 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved1 | FLCparamMB_FR_RES01 | REAL | 12381 | 12382 | 305D | 305E | | | | |
| 58 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved2 | FLCparamMB_FR_RES02 | REAL | 12383 | 12384 | 305F | 3060 | | | | |
| 59 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved3 | FLCparamMB_FR_RES03 | REAL | 12385 | 12386 | 3061 | 3062 | | | | |

| | | | | | | | | | | | | | | | | |
|-----|-----|----------------------------|----------------------------|---------------------------|----------------------------------|------|-------|-------|------|------|---------|-----|---------|--|-------|--|
| 60 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved4 | FLCparamMB_FR_RES04 | REAL | 12387 | 12388 | 3063 | 3064 | | | | | | |
| 61 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved5 | FLCparamMB_FR_RES05 | REAL | 12389 | 12390 | 3065 | 3066 | | | | | | |
| 62 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved6 | FLCparamMB_FR_RES06 | REAL | 12391 | 12392 | 3067 | 3068 | | | | | | |
| 63 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved7 | FLCparamMB_FR_RES07 | REAL | 12393 | 12394 | 3069 | 306A | | | | | | |
| 64 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved8 | FLCparamMB_FR_RES08 | REAL | 12395 | 12396 | 306B | 306C | | | | | | |
| 65 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved9 | FLCparamMB_FR_RES09 | REAL | 12397 | 12398 | 306D | 306E | | | | | | |
| 66 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved10 | FLCparamMB_FR_RES10 | REAL | 12399 | 12400 | 306F | 3070 | | | | | | |
| 67 | FLC | EtherCAT_Parameters_Modbus | FrequencyRegulation_Params | Reserved11 | FLCparamMB_FR_RES11 | REAL | 12401 | 12402 | 3071 | 3072 | | | | | | |
| 68 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | ModalFreq_Gain | FLCparamMB_MFA_Gain | REAL | 12403 | 12404 | 3073 | 3074 | 2000 | 0 | 1000 | | Gain | |
| 69 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | BandPassFilter_CenterFreq | FLCparamMB_MFA_BPF_CenterFreq | REAL | 12405 | 12406 | 3075 | 3076 | 50 | 0 | 31,4 | | rad/s | |
| 70 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | PowerLimit_Modalfreq | FLCparamMB_MFA_PowerLimit_MF | REAL | 12407 | 12408 | 3077 | 3078 | 1000000 | 0 | 1000000 | | Watts | |
| 71 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved1 | FLCparamMB_MFA_RES01 | REAL | 12409 | 12410 | 3079 | 307A | | | | | | |
| 72 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved2 | FLCparamMB_MFA_RES02 | REAL | 12411 | 12412 | 307B | 307C | | | | | | |
| 73 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved3 | FLCparamMB_MFA_RES03 | REAL | 12413 | 12414 | 307D | 307E | | | | | | |
| 74 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved4 | FLCparamMB_MFA_RES04 | REAL | 12415 | 12416 | 307F | 3080 | | | | | | |
| 75 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved5 | FLCparamMB_MFA_RES05 | REAL | 12417 | 12418 | 3081 | 3082 | | | | | | |
| 76 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved6 | FLCparamMB_MFA_RES06 | REAL | 12419 | 12420 | 3083 | 3084 | | | | | | |
| 77 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved7 | FLCparamMB_MFA_RES07 | REAL | 12421 | 12422 | 3085 | 3086 | | | | | | |
| 78 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved8 | FLCparamMB_MFA_RES08 | REAL | 12423 | 12424 | 3087 | 3088 | | | | | | |
| 79 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved9 | FLCparamMB_MFA_RES09 | REAL | 12425 | 12426 | 3089 | 308A | | | | | | |
| 80 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved10 | FLCparamMB_MFA_RES10 | REAL | 12427 | 12428 | 308B | 308C | | | | | | |
| 81 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved11 | FLCparamMB_MFA_RES11 | REAL | 12429 | 12430 | 308D | 308E | | | | | | |
| 82 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved12 | FLCparamMB_MFA_RES12 | REAL | 12431 | 12432 | 308F | 3090 | | | | | | |
| 83 | FLC | EtherCAT_Parameters_Modbus | ModalFrequency_Params | Reserved13 | FLCparamMB_MFA_RES13 | REAL | 12433 | 12434 | 3091 | 3092 | | | | | | |
| 84 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | DroopConstantV | FLCparamMB_VR_DroopConstantV | REAL | 12435 | 12436 | 3093 | 3094 | 100 | 0 | 20 | | % | |
| 85 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | PowerLimit_Voltage | FLCparamMB_VR_PowerLimit_Voltage | REAL | 12437 | 12438 | 3095 | 3096 | 1000000 | 0 | 1000000 | | Var | |
| 86 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | VoltageCutOff | FLCparamMB_VR_VoltageCutOff | REAL | 12439 | 12440 | 3097 | 3098 | 50 | 0 | 0,1 | | Hz | |
| 87 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | VoltDeadBand | FLCparamMB_VR_VoltDeadBand | REAL | 12441 | 12442 | 3099 | 309A | 400 | 0 | 20 | | Volts | |
| 88 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved1 | FLCparamMB_VR_RES01 | REAL | 12443 | 12444 | 309B | 309C | | | | | | |
| 89 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved2 | FLCparamMB_VR_RES02 | REAL | 12445 | 12446 | 309D | 309E | | | | | | |
| 90 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved3 | FLCparamMB_VR_RES03 | REAL | 12447 | 12448 | 309F | 30A0 | | | | | | |
| 91 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved4 | FLCparamMB_VR_RES04 | REAL | 12449 | 12450 | 30A1 | 30A2 | | | | | | |
| 92 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved5 | FLCparamMB_VR_RES05 | REAL | 12451 | 12452 | 30A3 | 30A4 | | | | | | |
| 93 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved6 | FLCparamMB_VR_RES06 | REAL | 12453 | 12454 | 30A5 | 30A6 | | | | | | |
| 94 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved7 | FLCparamMB_VR_RES07 | REAL | 12455 | 12456 | 30A7 | 30A8 | | | | | | |
| 95 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved8 | FLCparamMB_VR_RES08 | REAL | 12457 | 12458 | 30A9 | 30AA | | | | | | |
| 96 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved9 | FLCparamMB_VR_RES09 | REAL | 12459 | 12460 | 30AB | 30AC | | | | | | |
| 97 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved10 | FLCparamMB_VR_RES10 | REAL | 12461 | 12462 | 30AD | 30AE | | | | | | |
| 98 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved11 | FLCparamMB_VR_RES11 | REAL | 12463 | 12464 | 30AF | 30B0 | | | | | | |
| 99 | FLC | EtherCAT_Parameters_Modbus | VoltageRegulation_Params | Reserved12 | FLCparamMB_VR_RES12 | REAL | 12465 | 12466 | 30B1 | 30B2 | | | | | | |
| 100 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved01 | FLCparamMB_AddServ_RES01 | REAL | 12467 | 12468 | 30B3 | 30B4 | | | | | | |
| 101 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved02 | FLCparamMB_AddServ_RES02 | REAL | 12469 | 12470 | 30B5 | 30B6 | | | | | | |
| 102 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved03 | FLCparamMB_AddServ_RES03 | REAL | 12471 | 12472 | 30B7 | 30B8 | | | | | | |
| 103 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved04 | FLCparamMB_AddServ_RES04 | REAL | 12473 | 12474 | 30B9 | 30BA | | | | | | |
| 104 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved05 | FLCparamMB_AddServ_RES05 | REAL | 12475 | 12476 | 30BB | 30BC | | | | | | |
| 105 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved06 | FLCparamMB_AddServ_RES06 | REAL | 12477 | 12478 | 30BD | 30BE | | | | | | |
| 106 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved07 | FLCparamMB_AddServ_RES07 | REAL | 12479 | 12480 | 30BF | 30C0 | | | | | | |
| 107 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved08 | FLCparamMB_AddServ_RES08 | REAL | 12481 | 12482 | 30C1 | 30C2 | | | | | | |
| 108 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved09 | FLCparamMB_AddServ_RES09 | REAL | 12483 | 12484 | 30C3 | 30C4 | | | | | | |
| 109 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved10 | FLCparamMB_AddServ_RES10 | REAL | 12485 | 12486 | 30C5 | 30C6 | | | | | | |
| 110 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved11 | FLCparamMB_AddServ_RES11 | REAL | 12487 | 12488 | 30C7 | 30C8 | | | | | | |
| 111 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved12 | FLCparamMB_AddServ_RES12 | REAL | 12489 | 12490 | 30C9 | 30CA | | | | | | |
| 112 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved13 | FLCparamMB_AddServ_RES13 | REAL | 12491 | 12492 | 30CB | 30CC | | | | | | |
| 113 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved14 | FLCparamMB_AddServ_RES14 | REAL | 12493 | 12494 | 30CD | 30CE | | | | | | |
| 114 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved15 | FLCparamMB_AddServ_RES15 | REAL | 12495 | 12496 | 30CF | 30D0 | | | | | | |
| 115 | FLC | EtherCAT_Parameters_Modbus | AdditionalService_Params | Reserved16 | FLCparamMB_AddServ_RES16 | REAL | 12497 | 12498 | 30D1 | 30D2 | | | | | | |
| 116 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | NominalPower | FLCparamEAP_COMM_NominalPower | REAL | 12499 | 12500 | 30D3 | 30D4 | 1000000 | 0 | 1000000 | | Watts | |
| 117 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | NominalFreq | FLCparamEAP_COMM_NominalFreq | REAL | 12501 | 12502 | 30D5 | 30D6 | 60 | 0 | 50 | | Hz | |
| 118 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | NominalVolt | FLCparamEAP_COMM_NominalVolt | REAL | 12503 | 12504 | 30D7 | 30D8 | 400 | 0 | 240 | | V | |
| 119 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | ActivePowerLimit | FLCparamEAP_COMM_ActivePowLim | REAL | 12505 | 12506 | 30D9 | 30DA | 1000000 | 0 | 1000000 | | Watts | |
| 120 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | ReactivePowerLimit | FLCparamEAP_COMM_ReactivePowLim | REAL | 12507 | 12508 | 30DB | 30DC | 1000000 | 0 | 1000000 | | Watts | |
| 121 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | Delta_limits | FLCparamEAP_COMM_Delta_limits | REAL | 12509 | 12510 | 30DD | 30DE | 10 | -10 | 1 | | | |

| | | | | | | | | | | | | | | |
|-----|-----|-------------------------|--------------------------------|---------------------------|----------------------------------|------|-------|-------|------|------|---------|-----|---------|---------|
| 122 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | E_Limits | FLCparamEAP_COMM_E_Limits | REAL | 12511 | 12512 | 30DF | 30E0 | 500 | 0 | 200 | V |
| 123 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | Kp_ReactiveControl | FLCparamEAP_COMM_Kp_ReactiveCtrl | REAL | 12513 | 12514 | 30E1 | 30E2 | 10 | -10 | 0,0001 | |
| 124 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | Ki_ReactiveControl | FLCparamEAP_COMM_Ki_ReactiveCtrl | REAL | 12515 | 12516 | 30E3 | 30E4 | 10 | -10 | 0,0001 | |
| 125 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | VirtualReactance | FLCparamEAP_COMM_VirtReactance | REAL | 12517 | 12518 | 30E5 | 30E6 | 1 | 0 | 0,3 | p.u |
| 126 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | VirtualResistor | FLCparamEAP_COMM_VirtResistor | REAL | 12519 | 12520 | 30E7 | 30E8 | 1 | 0 | 0,1 | p.u |
| 127 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | Reserved1 | FLCparamEAP_COMM_RES01 | REAL | 12521 | 12522 | 30E9 | 30EA | | | | |
| 128 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | Reserved2 | FLCparamEAP_COMM_RES02 | REAL | 12523 | 12524 | 30EB | 30EC | | | | |
| 129 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | Reserved3 | FLCparamEAP_COMM_RES03 | REAL | 12525 | 12526 | 30ED | 30EE | | | | |
| 130 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | Reserved4 | FLCparamEAP_COMM_RES04 | REAL | 12527 | 12528 | 30EF | 30F0 | | | | |
| 131 | FLC | EtherCAT_Parameters_EAP | CommonBlock_Params_EAP | Reserved5 | FLCparamEAP_COMM_RES05 | REAL | 12529 | 12530 | 30F1 | 30F2 | | | | |
| 132 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Inertia_Constant | FLCparamEAP_IE_Inertia_Constant | REAL | 12531 | 12532 | 30F3 | 30F4 | 20 | 0 | 5 | s |
| 133 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Damping_Constant | FLCparamEAP_IE_Damping_Constant | REAL | 12533 | 12534 | 30F5 | 30F6 | 10 | 0 | 0,7 | damping |
| 134 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | HighPassFilterConst | FLCparamEAP_IE_HighPassFiltConst | REAL | 12535 | 12536 | 30F7 | 30F8 | 10 | 0 | 0,1 | Hz |
| 135 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Max_ROCOF | FLCparamEAP_IE_Max_ROCOF | REAL | 12537 | 12538 | 30F9 | 30FA | 10 | 0 | 6 | MW/s |
| 136 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved1 | FLCparamEAP_IE_RES01 | REAL | 12539 | 12540 | 30FB | 30FC | | | | |
| 137 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved2 | FLCparamEAP_IE_RES02 | REAL | 12541 | 12542 | 30FD | 30FE | | | | |
| 138 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved3 | FLCparamEAP_IE_RES03 | REAL | 12543 | 12544 | 30FF | 3100 | | | | |
| 139 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved4 | FLCparamEAP_IE_RES04 | REAL | 12545 | 12546 | 3101 | 3102 | | | | |
| 140 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved5 | FLCparamEAP_IE_RES05 | REAL | 12547 | 12548 | 3103 | 3104 | | | | |
| 141 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved6 | FLCparamEAP_IE_RES06 | REAL | 12549 | 12550 | 3105 | 3106 | | | | |
| 142 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved7 | FLCparamEAP_IE_RES07 | REAL | 12551 | 12552 | 3107 | 3108 | | | | |
| 143 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved8 | FLCparamEAP_IE_RES08 | REAL | 12553 | 12554 | 3109 | 310A | | | | |
| 144 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved9 | FLCparamEAP_IE_RES09 | REAL | 12555 | 12556 | 310B | 310C | | | | |
| 145 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved10 | FLCparamEAP_IE_RES10 | REAL | 12557 | 12558 | 310D | 310E | | | | |
| 146 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved11 | FLCparamEAP_IE_RES11 | REAL | 12559 | 12560 | 310F | 3110 | | | | |
| 147 | FLC | EtherCAT_Parameters_EAP | InertiaEmulation_Params_EAP | Reserved12 | FLCparamEAP_IE_RES12 | REAL | 12561 | 12562 | 3111 | 3112 | | | | |
| 148 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | DroopConstantP | FLCparamEAP_FR_DroopConstantP | REAL | 12563 | 12564 | 3113 | 3114 | 100 | 0 | 4 | % |
| 149 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Max_Delta_Energy | FLCparamEAP_FR_Max_Delta_Energy | REAL | 12565 | 12566 | 3115 | 3116 | 1000000 | 0 | 1000000 | Watts |
| 150 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | PowerLimit_Freq | FLCparamEAP_FR_PowerLimit_Freq | REAL | 12567 | 12568 | 3117 | 3118 | 1000000 | 0 | 1000000 | Watts |
| 151 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | LowPassFilter_Cutoff_Freq | FLCparamEAP_FR_LPF_Cutoff_Freq | REAL | 12569 | 12570 | 3119 | 311A | 100 | 0 | 1 | Hz |
| 152 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | DeadBand_Limits_P | FLCparamEAP_FR_DeadBand_Limits_P | REAL | 12571 | 12572 | 311B | 311C | 1000000 | 0 | 1000000 | Watts |
| 153 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved1 | FLCparamEAP_FR_RES01 | REAL | 12573 | 12574 | 311D | 311E | | | | |
| 154 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved2 | FLCparamEAP_FR_RES02 | REAL | 12575 | 12576 | 311F | 3120 | | | | |
| 155 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved3 | FLCparamEAP_FR_RES03 | REAL | 12577 | 12578 | 3121 | 3122 | | | | |
| 156 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved4 | FLCparamEAP_FR_RES04 | REAL | 12579 | 12580 | 3123 | 3124 | | | | |
| 157 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved5 | FLCparamEAP_FR_RES05 | REAL | 12581 | 12582 | 3125 | 3126 | | | | |
| 158 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved6 | FLCparamEAP_FR_RES06 | REAL | 12583 | 12584 | 3127 | 3128 | | | | |
| 159 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved7 | FLCparamEAP_FR_RES07 | REAL | 12585 | 12586 | 3129 | 312A | | | | |
| 160 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved8 | FLCparamEAP_FR_RES08 | REAL | 12587 | 12588 | 312B | 312C | | | | |
| 161 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved9 | FLCparamEAP_FR_RES09 | REAL | 12589 | 12590 | 312D | 312E | | | | |
| 162 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved10 | FLCparamEAP_FR_RES10 | REAL | 12591 | 12592 | 312F | 3130 | | | | |
| 163 | FLC | EtherCAT_Parameters_EAP | FrequencyRegulation_Params_EAP | Reserved11 | FLCparamEAP_FR_RES11 | REAL | 12593 | 12594 | 3131 | 3132 | | | | |
| 164 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | ModalFreq_Gain | FLCparamEAP_MFA_Gain | REAL | 12595 | 12596 | 3133 | 3134 | 2000 | 0 | 1000 | |
| 165 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | ModalFreq_PowerLimit | FLCparamEAP_MFA_PowerLimit | REAL | 12597 | 12598 | 3135 | 3136 | 1000000 | 0 | 1000000 | Watts |
| 166 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Max_ROCOF_Modal | FLCparamEAP_MFA_Max_ROCOF_Modal | REAL | 12599 | 12600 | 3137 | 3138 | 100 | 0 | 5 | MW/s |
| 167 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | BandPassFilter_CenterFreq | FLCparamEAP_MFA_BPF_CenterFreq | REAL | 12601 | 12602 | 3139 | 313A | 10 | 0 | 31,4 | rad/s |
| 168 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved1 | FLCparamEAP_MFA_RES01 | REAL | 12603 | 12604 | 313B | 313C | | | | |
| 169 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved2 | FLCparamEAP_MFA_RES02 | REAL | 12605 | 12606 | 313D | 313E | | | | |
| 170 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved3 | FLCparamEAP_MFA_RES03 | REAL | 12607 | 12608 | 313F | 3140 | | | | |
| 171 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved4 | FLCparamEAP_MFA_RES04 | REAL | 12609 | 12610 | 3141 | 3142 | | | | |
| 172 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved5 | FLCparamEAP_MFA_RES05 | REAL | 12611 | 12612 | 3143 | 3144 | | | | |
| 173 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved6 | FLCparamEAP_MFA_RES06 | REAL | 12613 | 12614 | 3145 | 3146 | | | | |
| 174 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved7 | FLCparamEAP_MFA_RES07 | REAL | 12615 | 12616 | 3147 | 3148 | | | | |
| 175 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved8 | FLCparamEAP_MFA_RES08 | REAL | 12617 | 12618 | 3149 | 314A | | | | |
| 176 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved9 | FLCparamEAP_MFA_RES09 | REAL | 12619 | 12620 | 314B | 314C | | | | |
| 177 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved10 | FLCparamEAP_MFA_RES10 | REAL | 12621 | 12622 | 314D | 314E | | | | |
| 178 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved11 | FLCparamEAP_MFA_RES11 | REAL | 12623 | 12624 | 314F | 3150 | | | | |
| 179 | FLC | EtherCAT_Parameters_EAP | ModalFrequency_Params_EAP | Reserved12 | FLCparamEAP_MFA_RES12 | REAL | 12625 | 12626 | 3151 | 3152 | | | | |
| 180 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | DroopConstantQ | FLCparamEAP_VR_DroopConstantQ | REAL | 12627 | 12628 | 3153 | 3154 | 100 | 0 | 5 | |
| 181 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | LowPassFilter_Cutoff_Volt | FLCparamEAP_VR_LPF_Cutoff_Volt | REAL | 12629 | 12630 | 3155 | 3156 | 50 | 0 | 5 | |
| 182 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | DeadBand_Limit_Q | FLCparamEAP_VR_DeadBand_Limit_Q | REAL | 12631 | 12632 | 3157 | 3158 | 400 | 0 | 20 | |

| | | | | | | | | | | | | | | |
|-----|-----|-------------------------|-------------------------------|---------------------|--------------------------------|------|-------|-------|------|------|---------|-----|---------|----|
| 183 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | PowerLimit_Volt | FLCparamEAP_VR_PowerLimit_Volt | REAL | 12633 | 12634 | 3159 | 315A | 1000000 | 0 | 1000000 | |
| 184 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved1 | FLCparamEAP_VR_RES01 | REAL | 12635 | 12636 | 3158 | 315C | | | | |
| 185 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved2 | FLCparamEAP_VR_RES02 | REAL | 12637 | 12638 | 3150 | 315E | | | | |
| 186 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved3 | FLCparamEAP_VR_RES03 | REAL | 12639 | 12640 | 315F | 3160 | | | | |
| 187 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved4 | FLCparamEAP_VR_RES04 | REAL | 12641 | 12642 | 3161 | 3162 | | | | |
| 188 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved5 | FLCparamEAP_VR_RES05 | REAL | 12643 | 12644 | 3163 | 3164 | | | | |
| 189 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved6 | FLCparamEAP_VR_RES06 | REAL | 12645 | 12646 | 3165 | 3166 | | | | |
| 190 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved7 | FLCparamEAP_VR_RES07 | REAL | 12647 | 12648 | 3167 | 3168 | | | | |
| 191 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved8 | FLCparamEAP_VR_RES08 | REAL | 12649 | 12650 | 3169 | 316A | | | | |
| 192 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved9 | FLCparamEAP_VR_RES09 | REAL | 12651 | 12652 | 316B | 316C | | | | |
| 193 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved10 | FLCparamEAP_VR_RES10 | REAL | 12653 | 12654 | 316D | 316E | | | | |
| 194 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved11 | FLCparamEAP_VR_RES11 | REAL | 12655 | 12656 | 316F | 3170 | | | | |
| 195 | FLC | EtherCAT_Parameters_EAP | VoltageRegulation_Params_EAP | Reserved12 | FLCparamEAP_VR_RES12 | REAL | 12657 | 12658 | 3171 | 3172 | | | | |
| 196 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved01 | FLCparamEAP_AddServ_RES01 | REAL | 12659 | 12660 | 3173 | 3174 | | | | |
| 197 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved02 | FLCparamEAP_AddServ_RES02 | REAL | 12661 | 12662 | 3175 | 3176 | | | | |
| 198 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved03 | FLCparamEAP_AddServ_RES03 | REAL | 12663 | 12664 | 3177 | 3178 | | | | |
| 199 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved04 | FLCparamEAP_AddServ_RES04 | REAL | 12665 | 12666 | 3179 | 317A | | | | |
| 200 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved05 | FLCparamEAP_AddServ_RES05 | REAL | 12667 | 12668 | 317B | 317C | | | | |
| 201 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved06 | FLCparamEAP_AddServ_RES06 | REAL | 12669 | 12670 | 317D | 317E | | | | |
| 202 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved07 | FLCparamEAP_AddServ_RES07 | REAL | 12671 | 12672 | 317F | 3180 | | | | |
| 203 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved08 | FLCparamEAP_AddServ_RES08 | REAL | 12673 | 12674 | 3181 | 3182 | | | | |
| 204 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved09 | FLCparamEAP_AddServ_RES09 | REAL | 12675 | 12676 | 3183 | 3184 | | | | |
| 205 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved10 | FLCparamEAP_AddServ_RES10 | REAL | 12677 | 12678 | 3185 | 3186 | | | | |
| 206 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved11 | FLCparamEAP_AddServ_RES11 | REAL | 12679 | 12680 | 3187 | 3188 | | | | |
| 207 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved12 | FLCparamEAP_AddServ_RES12 | REAL | 12681 | 12682 | 3189 | 318A | | | | |
| 208 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved13 | FLCparamEAP_AddServ_RES13 | REAL | 12683 | 12684 | 318B | 318C | | | | |
| 209 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved14 | FLCparamEAP_AddServ_RES14 | REAL | 12685 | 12686 | 318D | 318E | | | | |
| 210 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved15 | FLCparamEAP_AddServ_RES15 | REAL | 12687 | 12688 | 318F | 3190 | | | | |
| 211 | FLC | EtherCAT_Parameters_EAP | Additional_Service_Params_EAP | Reserved16 | FLCparamEAP_AddServ_RES16 | REAL | 12689 | 12690 | 3191 | 3192 | | | | |
| 212 | SLC | SecondLevel_Parameters | CommonParameters_P | BattSoC | SLCparam_COMM_BattSoC | REAL | 12691 | 12692 | 3193 | 3194 | 100 | 0 | | |
| 213 | SLC | SecondLevel_Parameters | CommonParameters_P | BattMaxP | SLCparam_COMM_BattMaxP | REAL | 12693 | 12694 | 3195 | 3196 | 1 | 0 | 1 | MW |
| 214 | SLC | SecondLevel_Parameters | CommonParameters_P | BattMinP | SLCparam_COMM_BattMinP | REAL | 12695 | 12696 | 3197 | 3198 | 1 | 0 | 1 | MW |
| 215 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved1 | SLCparam_COMM_RES01 | REAL | 12697 | 12698 | 3199 | 319A | | | | |
| 216 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved2 | SLCparam_COMM_RES02 | REAL | 12699 | 12700 | 319B | 319C | | | | |
| 217 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved3 | SLCparam_COMM_RES03 | REAL | 12701 | 12702 | 319D | 319E | | | | |
| 218 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved4 | SLCparam_COMM_RES04 | REAL | 12703 | 12704 | 319F | 31A0 | | | | |
| 219 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved5 | SLCparam_COMM_RES05 | REAL | 12705 | 12706 | 31A1 | 31A2 | | | | |
| 220 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved6 | SLCparam_COMM_RES06 | REAL | 12707 | 12708 | 31A3 | 31A4 | | | | |
| 221 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved7 | SLCparam_COMM_RES07 | REAL | 12709 | 12710 | 31A5 | 31A6 | | | | |
| 222 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved8 | SLCparam_COMM_RES08 | REAL | 12711 | 12712 | 31A7 | 31A8 | | | | |
| 223 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved9 | SLCparam_COMM_RES09 | REAL | 12713 | 12714 | 31A9 | 31AA | | | | |
| 224 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved10 | SLCparam_COMM_RES10 | REAL | 12715 | 12716 | 31AB | 31AC | | | | |
| 225 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved11 | SLCparam_COMM_RES11 | REAL | 12717 | 12718 | 31AD | 31AE | | | | |
| 226 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved12 | SLCparam_COMM_RES12 | REAL | 12719 | 12720 | 31AF | 31B0 | | | | |
| 227 | SLC | SecondLevel_Parameters | CommonParameters_P | Reserved13 | SLCparam_COMM_RES13 | REAL | 12721 | 12722 | 31B1 | 31B2 | | | | |
| 228 | SLC | SecondLevel_Parameters | ResourceVariability_P | KP_RVM | SLCparam_RVM_KP | REAL | 12723 | 12724 | 31B3 | 31B4 | 10 | -10 | | |
| 229 | SLC | SecondLevel_Parameters | ResourceVariability_P | KI_RVM | SLCparam_RVM_KI | REAL | 12725 | 12726 | 31B5 | 31B6 | 10 | -10 | | |
| 230 | SLC | SecondLevel_Parameters | ResourceVariability_P | MaxRate_RVM | SLCparam_RVM_MaxRate | REAL | 12727 | 12728 | 31B7 | 31B8 | 1 | 0 | | |
| 231 | SLC | SecondLevel_Parameters | ResourceVariability_P | MaxDeviation_RVM | SLCparam_RVM_MaxDeviation | REAL | 12729 | 12730 | 31B9 | 31BA | 1 | 0 | | |
| 232 | SLC | SecondLevel_Parameters | ResourceVariability_P | MaxReconnection_RVM | SLCparam_RVM_MaxReconnection | REAL | 12731 | 12732 | 31BB | 31BC | 1 | 0 | | |
| 233 | SLC | SecondLevel_Parameters | ResourceVariability_P | Ts_RVM | SLCparam_RVM_Ts | REAL | 12733 | 12734 | 31BD | 31BE | 1 | 0 | | |
| 234 | SLC | SecondLevel_Parameters | ResourceVariability_P | LF_LPF_RVM | SLCparam_RVM_LF_LPF | REAL | 12735 | 12736 | 31BF | 31C0 | 50 | 0 | | |
| 235 | SLC | SecondLevel_Parameters | ResourceVariability_P | HF_HPF_RVM | SLCparam_RVM_HF_HPF | REAL | 12737 | 12738 | 31C1 | 31C2 | 5000 | 0 | | |
| 236 | SLC | SecondLevel_Parameters | ResourceVariability_P | RVM_Limit_Max | SLCparam_RVM_Limit_Max | REAL | 12739 | 12740 | 31C3 | 31C4 | 1 | 0 | | |
| 237 | SLC | SecondLevel_Parameters | ResourceVariability_P | RVM_Limit_Min | SLCparam_RVM_Limit_Min | REAL | 12741 | 12742 | 31C5 | 31C6 | 1 | 0 | | |
| 238 | SLC | SecondLevel_Parameters | ResourceVariability_P | Reserved3 | SLCparam_RVM_RES03 | REAL | 12743 | 12744 | 31C7 | 31C8 | | | | |
| 239 | SLC | SecondLevel_Parameters | ResourceVariability_P | Reserved4 | SLCparam_RVM_RES04 | REAL | 12745 | 12746 | 31C9 | 31CA | | | | |
| 240 | SLC | SecondLevel_Parameters | ResourceVariability_P | Reserved5 | SLCparam_RVM_RES05 | REAL | 12747 | 12748 | 31CB | 31CC | | | | |
| 241 | SLC | SecondLevel_Parameters | ResourceVariability_P | Reserved6 | SLCparam_RVM_RES06 | REAL | 12749 | 12750 | 31CD | 31CE | | | | |
| 242 | SLC | SecondLevel_Parameters | ResourceVariability_P | Reserved7 | SLCparam_RVM_RES07 | REAL | 12751 | 12752 | 31CF | 31D0 | | | | |
| 243 | SLC | SecondLevel_Parameters | ResourceVariability_P | Reserved8 | SLCparam_RVM_RES08 | REAL | 12753 | 12754 | 31D1 | 31D2 | | | | |
| 244 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | KP_PSM | SLCparam_PSM_KP | REAL | 12755 | 12756 | 31D3 | 31D4 | 10 | -10 | | |

| | | | | | | | | | | | | | | |
|-----|-----|------------------------|-----------------------|--------------------|---------------------------------|------|-------|-------|------|------|-------|-----|--|--|
| 245 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | KI_PSM | SLCparam_PSM_KI | REAL | 12757 | 12758 | 31D5 | 31D6 | 10 | -10 | | |
| 246 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | InputRef_PSM | SLCparam_PSM_InputRef | REAL | 12759 | 12760 | 31D7 | 31D8 | 1 | -1 | | |
| 247 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Ts_PSM | SLCparam_PSM_Ts | REAL | 12761 | 12762 | 31D9 | 31DA | 1 | 0 | | |
| 248 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | PSM_Limit_Max | SLCparam_PSM_Limit_Max | REAL | 12763 | 12764 | 31DB | 31DC | 1 | 0 | | |
| 249 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | PSM_Limit_Min | SLCparam_PSM_Limit_Min | REAL | 12765 | 12766 | 31DD | 31DE | 1 | 0 | | |
| 250 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved3 | SLCparam_PSM_RES03 | REAL | 12767 | 12768 | 31DF | 31E0 | | | | |
| 251 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved4 | SLCparam_PSM_RES04 | REAL | 12769 | 12770 | 31E1 | 31E2 | | | | |
| 252 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved5 | SLCparam_PSM_RES05 | REAL | 12771 | 12772 | 31E3 | 31E4 | | | | |
| 253 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved6 | SLCparam_PSM_RES06 | REAL | 12773 | 12774 | 31E5 | 31E6 | | | | |
| 254 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved7 | SLCparam_PSM_RES07 | REAL | 12775 | 12776 | 31E7 | 31E8 | | | | |
| 255 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved8 | SLCparam_PSM_RES08 | REAL | 12777 | 12778 | 31E9 | 31EA | | | | |
| 256 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved9 | SLCparam_PSM_RES09 | REAL | 12779 | 12780 | 31EB | 31EC | | | | |
| 257 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved10 | SLCparam_PSM_RES10 | REAL | 12781 | 12782 | 31ED | 31EE | | | | |
| 258 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved11 | SLCparam_PSM_RES11 | REAL | 12783 | 12784 | 31EF | 31FO | | | | |
| 259 | SLC | SecondLevel_Parameters | PowerShiftingManage_P | Reserved12 | SLCparam_PSM_RES12 | REAL | 12785 | 12786 | 31F1 | 31F2 | | | | |
| 260 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | KP_RRE | SLCparam_RRE_KP | REAL | 12787 | 12788 | 31F3 | 31F4 | 10 | -10 | | |
| 261 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | KI_RRE | SLCparam_RRE_KI | REAL | 12789 | 12790 | 31F5 | 31F6 | 10 | -10 | | |
| 262 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | MaxRate_RRE | SLCparam_RRE_MaxRate | REAL | 12791 | 12792 | 31F7 | 31F8 | 1 | 0 | | |
| 263 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | RateLimited_RRE | SLCparam_RRE_RateLimited | REAL | 12793 | 12794 | 31F9 | 31FA | 1 | 0 | | |
| 264 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | Ts_RRE | SLCparam_RRE_Ts | REAL | 12795 | 12796 | 31FB | 31FC | 1 | 0 | | |
| 265 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | SampligFrecuency | SLCparam_RRE_SamplingFrecuency | REAL | 12797 | 12798 | 31FD | 31FE | 10000 | 0 | | |
| 266 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | RRE_Limit_Max | SLCparam_RRE_Limit_Max | REAL | 12799 | 12800 | 31FF | 3200 | 1 | 0 | | |
| 267 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | RRE_Limit_Min | SLCparam_RRE_Limit_Min | REAL | 12801 | 12802 | 3201 | 3202 | 1 | 0 | | |
| 268 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | Reserved3 | SLCparam_RRE_RES03 | REAL | 12803 | 12804 | 3203 | 3204 | | | | |
| 269 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | Reserved4 | SLCparam_RRE_RES04 | REAL | 12805 | 12806 | 3205 | 3206 | | | | |
| 270 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | Reserved5 | SLCparam_RRE_RES05 | REAL | 12807 | 12808 | 3207 | 3208 | | | | |
| 271 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | Reserved6 | SLCparam_RRE_RES06 | REAL | 12809 | 12810 | 3209 | 320A | | | | |
| 272 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | Reserved7 | SLCparam_RRE_RES07 | REAL | 12811 | 12812 | 320B | 320C | | | | |
| 273 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | Reserved8 | SLCparam_RRE_RES08 | REAL | 12813 | 12814 | 320D | 320E | | | | |
| 274 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | Reserved9 | SLCparam_RRE_RES09 | REAL | 12815 | 12816 | 320F | 3210 | | | | |
| 275 | SLC | SecondLevel_Parameters | RampRateEnhancemet_P | Reserved10 | SLCparam_RRE_RES10 | REAL | 12817 | 12818 | 3211 | 3212 | | | | |
| 276 | SLC | SecondLevel_Parameters | PowerCurtailment_P | KP_PCM | SLCparam_PCM_KP | REAL | 12819 | 12820 | 3213 | 3214 | 10 | -10 | | |
| 277 | SLC | SecondLevel_Parameters | PowerCurtailment_P | KI_PCM | SLCparam_PCM_KI | REAL | 12821 | 12822 | 3215 | 3216 | 10 | -10 | | |
| 278 | SLC | SecondLevel_Parameters | PowerCurtailment_P | BalanceControlRef | SLCparam_PCM_BalanceControlRef | REAL | 12823 | 12824 | 3217 | 3218 | 1 | 0 | | |
| 279 | SLC | SecondLevel_Parameters | PowerCurtailment_P | DeltaControlRef | SLCparam_PCM_DeltaControlRef | REAL | 12825 | 12826 | 3219 | 321A | 1 | 0 | | |
| 280 | SLC | SecondLevel_Parameters | PowerCurtailment_P | StopRegulationP | SLCparam_PCM_StopRegulationP | REAL | 12827 | 12828 | 321B | 321C | 1 | 0 | | |
| 281 | SLC | SecondLevel_Parameters | PowerCurtailment_P | Ts_PCM | SLCparam_PCM_Ts | REAL | 12829 | 12830 | 321D | 321E | 1 | 0 | | |
| 282 | SLC | SecondLevel_Parameters | PowerCurtailment_P | PCM_Limit_Max | SLCparam_PCM_Limit_Max | REAL | 12831 | 12832 | 321F | 3220 | 1 | 0 | | |
| 283 | SLC | SecondLevel_Parameters | PowerCurtailment_P | PCM_Limit_Min | SLCparam_PCM_Limit_Min | REAL | 12833 | 12834 | 3221 | 3222 | 1 | 0 | | |
| 284 | SLC | SecondLevel_Parameters | PowerCurtailment_P | Reserved3 | SLCparam_PCM_RES03 | REAL | 12835 | 12836 | 3223 | 3224 | | | | |
| 285 | SLC | SecondLevel_Parameters | PowerCurtailment_P | Reserved4 | SLCparam_PCM_RES04 | REAL | 12837 | 12838 | 3225 | 3226 | | | | |
| 286 | SLC | SecondLevel_Parameters | PowerCurtailment_P | Reserved5 | SLCparam_PCM_RES05 | REAL | 12839 | 12840 | 3227 | 3228 | | | | |
| 287 | SLC | SecondLevel_Parameters | PowerCurtailment_P | Reserved6 | SLCparam_PCM_RES06 | REAL | 12841 | 12842 | 3229 | 322A | | | | |
| 288 | SLC | SecondLevel_Parameters | PowerCurtailment_P | Reserved7 | SLCparam_PCM_RES07 | REAL | 12843 | 12844 | 322B | 322C | | | | |
| 289 | SLC | SecondLevel_Parameters | PowerCurtailment_P | Reserved8 | SLCparam_PCM_RES08 | REAL | 12845 | 12846 | 322D | 322E | | | | |
| 290 | SLC | SecondLevel_Parameters | PowerCurtailment_P | Reserved9 | SLCparam_PCM_RES09 | REAL | 12847 | 12848 | 322F | 3230 | | | | |
| 291 | SLC | SecondLevel_Parameters | PowerCurtailment_P | Reserved10 | SLCparam_PCM_RES10 | REAL | 12849 | 12850 | 3231 | 3232 | | | | |
| 292 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | KP_PBP | SLCparam_PBP_KP | REAL | 12851 | 12852 | 3233 | 3234 | 10 | -10 | | |
| 293 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | KI_PBP | SLCparam_PBP_KI | REAL | 12853 | 12854 | 3235 | 3236 | 10 | -10 | | |
| 294 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | LinePowerReference | SLCparam_PBP_LinePowerReference | REAL | 12855 | 12856 | 3237 | 3238 | 1 | -1 | | |
| 295 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | FrequencyReference | SLCparam_PBP_FrequencyReference | REAL | 12857 | 12858 | 3239 | 323A | 60 | 0 | | |
| 296 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Ts_PBP | SLCparam_PBP_Ts | REAL | 12859 | 12860 | 323B | 323C | 1 | 0 | | |
| 297 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | PBP_Limit_Max | SLCparam_PBP_Limit_Max | REAL | 12861 | 12862 | 323D | 323E | 1 | 0 | | |
| 298 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | PBP_Limit_Min | SLCparam_PBP_Limit_Min | REAL | 12863 | 12864 | 323F | 3240 | 1 | 0 | | |
| 299 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Reserved3 | SLCparam_PBP_RES03 | REAL | 12865 | 12866 | 3241 | 3242 | | | | |
| 300 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Reserved4 | SLCparam_PBP_RES04 | REAL | 12867 | 12868 | 3243 | 3244 | | | | |
| 301 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Reserved5 | SLCparam_PBP_RES05 | REAL | 12869 | 12870 | 3245 | 3246 | | | | |
| 302 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Reserved6 | SLCparam_PBP_RES06 | REAL | 12871 | 12872 | 3247 | 3248 | | | | |
| 303 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Reserved7 | SLCparam_PBP_RES07 | REAL | 12873 | 12874 | 3249 | 324A | | | | |
| 304 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Reserved8 | SLCparam_PBP_RES08 | REAL | 12875 | 12876 | 324B | 324C | | | | |
| 305 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Reserved9 | SLCparam_PBP_RES09 | REAL | 12877 | 12878 | 324D | 324E | | | | |
| 306 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Reserved10 | SLCparam_PBP_RES10 | REAL | 12879 | 12880 | 324F | 3250 | | | | |

| | | | | | | | | | | | | | | |
|-----|-----|------------------------|----------------------------|-------------------------------------|-------------------------------|------|-------|-------|------|------|-------|----|--|--|
| 307 | SLC | SecondLevel_Parameters | PowerBalanceProv_P | Reserved11 | SLCparam_PBP_RES11 | REAL | 12881 | 12882 | 3251 | 3252 | | | | |
| 308 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved01 | SLCparam_AddServ_RES01 | REAL | 12883 | 12884 | 3253 | 3254 | | | | |
| 309 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved02 | SLCparam_AddServ_RES02 | REAL | 12885 | 12886 | 3255 | 3256 | | | | |
| 310 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved03 | SLCparam_AddServ_RES03 | REAL | 12887 | 12888 | 3257 | 3258 | | | | |
| 311 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved04 | SLCparam_AddServ_RES04 | REAL | 12889 | 12890 | 3259 | 325A | | | | |
| 312 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved05 | SLCparam_AddServ_RES05 | REAL | 12891 | 12892 | 3258 | 325C | | | | |
| 313 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved06 | SLCparam_AddServ_RES06 | REAL | 12893 | 12894 | 3250 | 325E | | | | |
| 314 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved07 | SLCparam_AddServ_RES07 | REAL | 12895 | 12896 | 325F | 3260 | | | | |
| 315 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved08 | SLCparam_AddServ_RES08 | REAL | 12897 | 12898 | 3261 | 3262 | | | | |
| 316 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved09 | SLCparam_AddServ_RES09 | REAL | 12899 | 12900 | 3263 | 3264 | | | | |
| 317 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved10 | SLCparam_AddServ_RES10 | REAL | 12901 | 12902 | 3265 | 3266 | | | | |
| 318 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved11 | SLCparam_AddServ_RES11 | REAL | 12903 | 12904 | 3267 | 3268 | | | | |
| 319 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved12 | SLCparam_AddServ_RES12 | REAL | 12905 | 12906 | 3269 | 326A | | | | |
| 320 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved13 | SLCparam_AddServ_RES13 | REAL | 12907 | 12908 | 326B | 326C | | | | |
| 321 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved14 | SLCparam_AddServ_RES14 | REAL | 12909 | 12910 | 326D | 326E | | | | |
| 322 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved15 | SLCparam_AddServ_RES15 | REAL | 12911 | 12912 | 326F | 3270 | | | | |
| 323 | SLC | SecondLevel_Parameters | AdditionalService_P | Reserved16 | SLCparam_AddServ_RES16 | REAL | 12913 | 12914 | 3271 | 3272 | | | | |
| 324 | SLC | Inputs | SCADA_Inputs | Commands | FLCIn_input_Commands | DINT | 12915 | 12916 | 3273 | 3274 | | | | |
| 325 | SLC | Inputs | SCADA_Inputs | ActivePowerReference | FLCIn_input_ActivePowerRef | REAL | 12917 | 12918 | 3275 | 3276 | 1 | 0 | | |
| 326 | SLC | Inputs | SCADA_Inputs | ReactivePowerReference | FLCIn_input_ReactivePowerRef | REAL | 12919 | 12920 | 3277 | 3278 | 1 | 0 | | |
| 327 | SLC | Inputs | SCADA_Inputs | Reserved2 | FLCIn_input_RES02 | REAL | 12921 | 12922 | 3279 | 327A | | | | |
| 328 | SLC | Inputs | SCADA_Inputs | Reserved3 | FLCIn_input_RES03 | REAL | 12923 | 12924 | 327B | 327C | | | | |
| 329 | SLC | Inputs | SCADA_Inputs | Reserved4 | FLCIn_input_RES04 | REAL | 12925 | 12926 | 327D | 327E | | | | |
| 330 | SLC | Inputs | SCADA_Inputs | Reserved5 | FLCIn_input_RES05 | REAL | 12927 | 12928 | 327F | 3280 | | | | |
| 331 | SLC | Inputs | SCADA_Inputs | Reserved6 | FLCIn_input_RES06 | REAL | 12929 | 12930 | 3281 | 3282 | | | | |
| 332 | SLC | Inputs | SCADA_Inputs | Reserved7 | FLCIn_input_RES07 | REAL | 12931 | 12932 | 3283 | 3284 | | | | |
| 333 | SLC | Inputs | SCADA_Inputs | Reserved8 | FLCIn_input_RES08 | REAL | 12933 | 12934 | 3285 | 3286 | | | | |
| 334 | SLC | Inputs | SCADA_Inputs | Reserved9 | FLCIn_input_RES09 | REAL | 12935 | 12936 | 3287 | 3288 | | | | |
| 335 | SLC | Inputs | SCADA_Inputs | Reserved10 | FLCIn_input_RES10 | REAL | 12937 | 12938 | 3289 | 328A | | | | |
| 336 | SLC | Inputs | SCADA_Inputs | Reserved11 | FLCIn_input_RES11 | REAL | 12939 | 12940 | 328B | 328C | | | | |
| 337 | SLC | Inputs | SCADA_Inputs | Reserved12 | FLCIn_input_RES12 | REAL | 12941 | 12942 | 328D | 328E | | | | |
| 338 | SLC | Inputs | SCADA_Inputs | Reserved13 | FLCIn_input_RES13 | REAL | 12943 | 12944 | 328F | 3290 | | | | |
| 339 | SLC | Inputs | SCADA_Inputs | Reserved14 | FLCIn_input_RES14 | REAL | 12945 | 12946 | 3291 | 3292 | | | | |
| 340 | FLC | Outputs | EtherCAT_Stat_Variables | Status | FLCIn_statVar_Status | DINT | 12947 | 12948 | 3293 | 3294 | 6 | 0 | | |
| 341 | FLC | Outputs | EtherCAT_Stat_Variables | HeartBeat | FLCIn_statVar_HeartBeat | DINT | 12949 | 12950 | 3295 | 3296 | 65535 | 0 | | |
| 342 | FLC | Outputs | EtherCAT_Stat_Variables | Alarms1 | FLCIn_statVar_Alarms1 | DINT | 12951 | 12952 | 3297 | 3298 | 65535 | 0 | | |
| 343 | FLC | Outputs | EtherCAT_Stat_Variables | Alarms2 | FLCIn_statVar_Alarms2 | DINT | 12953 | 12954 | 3299 | 329A | 65535 | 0 | | |
| 344 | FLC | Outputs | EtherCAT_Stat_Variables | Alarms3 | FLCIn_statVar_Alarms3 | DINT | 12955 | 12956 | 329B | 329C | 65535 | 0 | | |
| 345 | FLC | Outputs | EtherCAT_Stat_Variables | Alarms4 | FLCIn_statVar_Alarms4 | DINT | 12957 | 12958 | 329D | 329E | 65535 | 0 | | |
| 346 | FLC | Outputs | EtherCAT_Stat_Variables | WarningLog | FLCIn_statVar_WarningLog | DINT | 12959 | 12960 | 329F | 32A0 | | | | |
| 347 | FLC | Outputs | EtherCAT_Stat_Variables | Reserved2 | FLCIn_statVar_RES02 | DINT | 12961 | 12962 | 32A1 | 32A2 | | | | |
| 348 | FLC | Outputs | EtherCAT_Monitor_Variables | BatSoc | FLCIn_MonitorVar_BatSoc | REAL | 12963 | 12964 | 32A3 | 32A4 | 100 | 0 | | |
| 349 | FLC | Outputs | EtherCAT_Monitor_Variables | BatSocUpLim | FLCIn_MonitorVar_BatSocUpLim | REAL | 12965 | 12966 | 32A5 | 32A6 | 90 | 51 | | |
| 350 | FLC | Outputs | EtherCAT_Monitor_Variables | BatSocLowLim | FLCIn_MonitorVar_BatSocLowLim | REAL | 12967 | 12968 | 32A7 | 32A8 | 50 | 10 | | |
| 351 | FLC | Outputs | EtherCAT_Monitor_Variables | MaxWArAval | FLCIn_MonitorVar_MaxWArAval | REAL | 12969 | 12970 | 32A9 | 32AA | 1 | 0 | | |
| 352 | FLC | Outputs | EtherCAT_Monitor_Variables | MaxWCharVal | FLCIn_MonitorVar_MaxWCharVal | REAL | 12971 | 12972 | 32AB | 32AC | 1 | 0 | | |
| 353 | FLC | Outputs | EtherCAT_Monitor_Variables | MaxVArAval | FLCIn_MonitorVar_MaxVArAval | REAL | 12973 | 12974 | 32AD | 32AE | 1 | 0 | | |
| 354 | FLC | Outputs | EtherCAT_Monitor_Variables | MinVArAval | FLCIn_MonitorVar_MinVArAval | REAL | 12975 | 12976 | 32AF | 32B0 | 1 | 0 | | |
| 355 | FLC | Outputs | EtherCAT_Monitor_Variables | TotW | FLCIn_MonitorVar_TotW | REAL | 12977 | 12978 | 32B1 | 32B2 | 1 | 0 | | |
| 356 | FLC | Outputs | EtherCAT_Monitor_Variables | TotVAr | FLCIn_MonitorVar_TotVAr | REAL | 12979 | 12980 | 32B3 | 32B4 | 1 | 0 | | |
| 357 | FLC | Outputs | EtherCAT_Monitor_Variables | GridFre | FLCIn_MonitorVar_GridFre | REAL | 12981 | 12982 | 32B5 | 32B6 | 60 | 0 | | |
| 358 | FLC | Outputs | EtherCAT_Monitor_Variables | Vrmsa | FLCIn_MonitorVar_Vrmsa | REAL | 12983 | 12984 | 32B7 | 32B8 | 240 | 0 | | |
| 359 | FLC | Outputs | EtherCAT_Monitor_Variables | Vrmsb | FLCIn_MonitorVar_Vrmsb | REAL | 12985 | 12986 | 32B9 | 32BA | 240 | 0 | | |
| 360 | FLC | Outputs | EtherCAT_Monitor_Variables | Vrmsc | FLCIn_MonitorVar_Vrmsc | REAL | 12987 | 12988 | 32BB | 32BC | 240 | 0 | | |
| 361 | FLC | Outputs | EtherCAT_Monitor_Variables | Irmsa | FLCIn_MonitorVar_Irmsa | REAL | 12989 | 12990 | 32BD | 32BE | 1500 | 0 | | |
| 362 | FLC | Outputs | EtherCAT_Monitor_Variables | Irmsb | FLCIn_MonitorVar_Irmsb | REAL | 12991 | 12992 | 32BF | 32C0 | 1500 | 0 | | |
| 363 | FLC | Outputs | EtherCAT_Monitor_Variables | Irmsc | FLCIn_MonitorVar_Irmsc | REAL | 12993 | 12994 | 32C1 | 32C2 | 1500 | 0 | | |
| 364 | FLC | Output | EtherCAT_Output_Modbus | Delta_Power_out_InertiaEmulation | FLCout_MB_DPo_IIE | REAL | 12995 | 12996 | 32C3 | 32C4 | 1 | 0 | | |
| 365 | FLC | Output | EtherCAT_Output_Modbus | Delta_Power_out_FrequencyRegulation | FLCout_MB_DPo_FR | REAL | 12997 | 12998 | 32C5 | 32C6 | 1 | 0 | | |
| 366 | FLC | Output | EtherCAT_Output_Modbus | Delta_Power_out_ModFreqAtten | FLCout_MB_DPo_MFA | REAL | 12999 | 13000 | 32C7 | 32C8 | 1 | 0 | | |
| 367 | FLC | Output | EtherCAT_Output_Modbus | Delta_Power_out_VoltageReg | FLCout_MB_DPo_VR | REAL | 13001 | 13002 | 32C9 | 32CA | 1 | 0 | | |
| 368 | FLC | Output | EtherCAT_Output_EAP | Delta_Power_out_InertiaEmulation | FLCout_EAP_DPo_IIE | REAL | 13003 | 13004 | 32CB | 32CC | 1 | 0 | | |

| | | | | | | | | | | | | | | |
|-----|-----|---------------------|-----------------------------------|-------------------------------------|--------------------|------|-------|-------|------|------|---|---|--|--|
| 369 | FLC | Output | EtherCAT_Output_EAP | Delta_Power_out_FrequencyRegulation | FLCout_EAP_DPo_FR | REAL | 13005 | 13006 | 32CD | 32CE | 1 | 0 | | |
| 370 | FLC | Output | EtherCAT_Output_EAP | Delta_Power_out_ModFreqAtten | FLCout_EAP_DPo_MFA | REAL | 13007 | 13008 | 32CF | 32D0 | 1 | 0 | | |
| 371 | FLC | Output | EtherCAT_Output_EAP | Delta_Power_out_VoltageReg | FLCout_EAP_DPo_VR | REAL | 13009 | 13010 | 32D1 | 32D2 | 1 | 0 | | |
| 372 | SLC | SecondLevel_Outputs | ResourceVariability_P | ActivePowerReference_RVM | SLCout_RVM | REAL | 13011 | 13012 | 32D3 | 32D4 | 1 | 0 | | |
| 373 | SLC | SecondLevel_Outputs | PowerShiftingManage_P | ActivePowerReference_PSM | SLCout_PSM | REAL | 13013 | 13014 | 32D5 | 32D6 | 1 | 0 | | |
| 374 | SLC | SecondLevel_Outputs | RampRateEnhancemet_P | ActivePowerReference_RRE | SLCout_RRE | REAL | 13015 | 13016 | 32D7 | 32D8 | 1 | 0 | | |
| 375 | SLC | SecondLevel_Outputs | PowerCurtaIlment_P | ActivePowerReference_PCM | SLCout_PCM | REAL | 13017 | 13018 | 32D9 | 32DA | 1 | 0 | | |
| 376 | SLC | SecondLevel_Outputs | PowerBalanceProv_P | ActivePowerReference_PBP | SLCout_PBP | REAL | 13019 | 13020 | 32DB | 32DC | 1 | 0 | | |
| 377 | SLC | SecondLevel_Outputs | CongestionRelief_P | ActivePowerReference_CR | SLCout_CR | REAL | 13021 | 13022 | 32DD | 32DE | 1 | 0 | | |
| 378 | TLC | ThirdLevel_Commands | ThirdLevel_Control | ThirdLevel_commands | TLCInput_Command | DINT | 13023 | 13024 | 32DF | 32E0 | | | | |
| 971 | AS | Output_AS | BusbarVoltage | B_Athienou_132kV_V | B_Athienou_132kV_V | REAL | 13658 | 13659 | 355A | 355B | | | | |
| 972 | AS | Output_AS | BusbarVoltage | B_Athienou_11kV1_V | B_Athienou_11kV1_V | REAL | 13660 | 13661 | 355C | 355D | | | | |
| 973 | AS | Output_AS | BusbarVoltage | B_Athienou_11kV2_V | B_Athienou_11kV2_V | REAL | 13662 | 13663 | 355E | 355F | | | | |
| 974 | AS | Output_AS | BusbarVoltage | B_BESS_V | B_BESS_V | REAL | 13664 | 13665 | 3560 | 3561 | | | | |
| 975 | AS | Output_AS | Busbar frequency | B_Athienou_132kV_f | B_Athienou_132kV_f | REAL | 13666 | 13667 | 3562 | 3563 | | | | |
| 976 | AS | Output_AS | Transformer_active power | TR1_P | TR1_P | REAL | 13668 | 13669 | 3564 | 3565 | | | | |
| 977 | AS | Output_AS | Transformer_reactive power | TR1_Q | TR1_Q | REAL | 13670 | 13671 | 3566 | 3567 | | | | |
| 978 | AS | Output_AS | Transformer_active power | TR2_P | TR2_P | REAL | 13672 | 13673 | 3568 | 3569 | | | | |
| 979 | AS | Output_AS | Transformer_reactive power | TR2_Q | TR2_Q | REAL | 13674 | 13675 | 356A | 356B | | | | |
| 980 | AS | Output_AS | Load_active power | LD_1_P | LD_1_P | REAL | 13676 | 13677 | 356C | 356D | | | | |
| 981 | AS | Output_AS | Load_reactive power | LD_1_Q | LD_1_Q | REAL | 13678 | 13679 | 356E | 356F | | | | |
| 982 | AS | Output_AS | Load_active power | LD_2_P | LD_2_P | REAL | 13680 | 13681 | 3570 | 3571 | | | | |
| 983 | AS | Output_AS | Load_reactive power | LD_2_Q | LD_2_Q | REAL | 13682 | 13683 | 3572 | 3573 | | | | |
| 984 | AS | Output_AS | Load_active power | LD_3_P | LD_3_P | REAL | 13684 | 13685 | 3574 | 3575 | | | | |
| 985 | AS | Output_AS | Load_reactive power | LD_3_Q | LD_3_Q | REAL | 13686 | 13687 | 3576 | 3577 | | | | |
| 986 | AS | Output_AS | Load_active power | LD_4_P | LD_4_P | REAL | 13688 | 13689 | 3578 | 3579 | | | | |
| 987 | AS | Output_AS | Load_reactive power | LD_4_Q | LD_4_Q | REAL | 13690 | 13691 | 357A | 357B | | | | |
| 988 | AS | Output_AS | BESS_active power | BESS_P | BESS_P | REAL | 13692 | 13693 | 357C | 357D | | | | |
| 989 | AS | Output_AS | BESS_reactive power | BESS_Q | BESS_Q | REAL | 13694 | 13695 | 357E | 357F | | | | |
| 990 | AS | Output_AS | Transformer_circuit braker_status | TR1_CB_HV | TR1_CB_HV | UINT | 13696 | | 3580 | 0 | | | | |
| 991 | AS | Output_AS | Transformer_circuit braker_status | TR1_CB_LV | TR1_CB_LV | UINT | 13697 | | 3581 | 0 | | | | |
| 992 | AS | Output_AS | Transformer_circuit braker_status | TR2_CB_HV | TR2_CB_HV | UINT | 13698 | | 3582 | 0 | | | | |
| 993 | AS | Output_AS | Transformer_circuit braker_status | TR2_CB_LV | TR2_CB_LV | UINT | 13699 | | 3583 | 0 | | | | |
| 994 | AS | Output_AS | Busbar circuit braker_status | B_11kV_CB | B_11kV_CB | UINT | 13700 | | 3584 | 0 | | | | |
| 995 | AS | Output_AS | Load_circuit braker_status | LD_1_CB | LD_1_CB | UINT | 13701 | | 3585 | 0 | | | | |
| 996 | AS | Output_AS | Load_circuit braker_status | LD_2_CB | LD_2_CB | UINT | 13702 | | 3586 | 0 | | | | |
| 997 | AS | Output_AS | Load_circuit braker_status | LD_3_CB | LD_3_CB | UINT | 13703 | | 3587 | 0 | | | | |
| 998 | AS | Output_AS | Load_circuit braker_status | LD_4_CB | LD_4_CB | UINT | 13704 | | 3588 | 0 | | | | |
| 999 | AS | Output_AS | BESS circuit braker_status | BESS_CB | BESS_CB | UINT | 13705 | | 3589 | 0 | | | | |

| INT16 | | | | 32 REAL | | | |
|---------------------------------------|------|-----|--|--------------------|------|----------|--|
| Communication - Modbus (Common Block) | | | | | | | |
| Comands name | Type | bit | | Parameter Name | Type | Position | |
| ActivePowerLimitation | Bool | 0 | | NominalPower | Real | 1 | |
| ReactivePowerLimitation | Bool | 1 | | NominalVoltage | Real | 2 | |
| State Of Health | Bool | 2 | | NominalFreq | Real | 3 | |
| Reset_To_Default_Common | Bool | 3 | | ActivePowerLimit | Real | 4 | |
| Reserved1 | Bool | 4 | | ReactivePowerLimit | Real | 5 | |
| Reserved2 | Bool | 5 | | Reserved1 | Real | 6 | |
| Reserved3 | Bool | 6 | | Reserved2 | Real | 7 | |
| Reserved4 | Bool | 7 | | Reserved3 | Real | 8 | |
| Reserved5 | Bool | 8 | | Reserved4 | Real | 9 | |
| Reserved6 | Bool | 9 | | Reserved5 | Real | 10 | |
| Reserved7 | Bool | 10 | | Reserved6 | Real | 11 | |
| Reserved8 | Bool | 11 | | Reserved7 | Real | 12 | |
| Reserved9 | Bool | 12 | | Reserved8 | Real | 13 | |
| Reserved10 | Bool | 13 | | Reserved9 | Real | 14 | |
| Reserved11 | Bool | 14 | | Reserved10 | Real | 15 | |
| Reserved12 | Bool | 15 | | Reserved11 | Real | 16 | |

| INT16 | | | | 32 REAL | | | |
|--|------|-----|--|---------------------------|------|----------|--|
| Communication - Modbus (Modal Frequency attenuation) | | | | | | | |
| Comands name | Type | bit | | Parameter Name | Type | Position | |
| Enable_ModalFreq | Bool | 0 | | ModalFreq_Gain | Real | 1 | |
| Delta_Power_Limit_ModalFreq | Bool | 1 | | BandPassFilter_CenterFreq | Real | 2 | |
| Reset_To_Default_ModalFreq | Bool | 2 | | PowerLimit_ModalFreq | Real | 3 | |
| Reserved1 | Bool | 3 | | Reserved1 | Real | 4 | |
| Reserved2 | Bool | 4 | | Reserved2 | Real | 5 | |
| Reserved3 | Bool | 5 | | Reserved3 | Real | 6 | |
| Reserved4 | Bool | 6 | | Reserved4 | Real | 7 | |
| Reserved5 | Bool | 7 | | Reserved5 | Real | 8 | |
| Reserved6 | Bool | 8 | | Reserved6 | Real | 9 | |
| Reserved7 | Bool | 9 | | Reserved7 | Real | 10 | |
| Reserved8 | Bool | 10 | | Reserved8 | Real | 11 | |
| Reserved9 | Bool | 11 | | Reserved9 | Real | 12 | |
| Reserved10 | Bool | 12 | | Reserved10 | Real | 13 | |
| Reserved11 | Bool | 13 | | Reserved11 | Real | 14 | |
| Reserved12 | Bool | 14 | | Reserved12 | Real | 15 | |
| Reserved13 | Bool | 15 | | Reserved13 | Real | 16 | |

| INT16 | | | | 32 REAL | | | |
|--|------|-----|--|---------------------|------|----------|--|
| Communication - Modbus (Inertia Emulation) | | | | | | | |
| Comands name | Type | bit | | Parameter Name | Type | Position | |
| Enable_Inertia | Bool | 0 | | Inertia_Constant | Real | 1 | |
| Inertia_Reset_States | Bool | 1 | | HighPassFilterConst | Real | 2 | |
| Limit_ROCOF | Bool | 2 | | Max_ROCOF | Real | 3 | |
| Delta_Power_Limit_Inertia | Bool | 3 | | PowerLimit_Inertia | Real | 4 | |
| Reset_To_Default_Inertia | Bool | 4 | | Reserved1 | Real | 5 | |
| Reserved1 | Bool | 5 | | Reserved2 | Real | 6 | |
| Reserved2 | Bool | 6 | | Reserved3 | Real | 7 | |
| Reserved3 | Bool | 7 | | Reserved4 | Real | 8 | |
| Reserved4 | Bool | 8 | | Reserved5 | Real | 9 | |
| Reserved5 | Bool | 9 | | Reserved6 | Real | 10 | |
| Reserved6 | Bool | 10 | | Reserved7 | Real | 11 | |
| Reserved7 | Bool | 11 | | Reserved8 | Real | 12 | |
| Reserved8 | Bool | 12 | | Reserved9 | Real | 13 | |
| Reserved9 | Bool | 13 | | Reserved10 | Real | 14 | |
| Reserved10 | Bool | 14 | | Reserved11 | Real | 15 | |
| Reserved11 | Bool | 15 | | Reserved12 | Real | 16 | |

| INT16 | | | | 32 REAL | | | |
|---|------|-----|--|--------------------|------|----------|--|
| Communication - Modbus (Voltage Regulation) | | | | | | | |
| Comands name | Type | bit | | Parameter Name | Type | Position | |
| Enable_VoltageRef | Bool | 0 | | DroopConstantV | Real | 1 | |
| Delta_Power_Limit_Voltage | Bool | 1 | | PowerLimit_Voltage | Real | 2 | |
| DeadBandLimitationV | Bool | 2 | | VoltageCutOff | Real | 3 | |
| Reset_To_Default_VoltageReg | Bool | 3 | | VoltDeadBand | Real | 4 | |
| Reserved1 | Bool | 4 | | Reserved1 | Real | 5 | |
| Reserved2 | Bool | 5 | | Reserved2 | Real | 6 | |
| Reserved3 | Bool | 6 | | Reserved3 | Real | 7 | |
| Reserved4 | Bool | 7 | | Reserved4 | Real | 8 | |
| Reserved5 | Bool | 8 | | Reserved5 | Real | 9 | |
| Reserved6 | Bool | 9 | | Reserved6 | Real | 10 | |
| Reserved7 | Bool | 10 | | Reserved7 | Real | 11 | |
| Reserved8 | Bool | 11 | | Reserved8 | Real | 12 | |
| Reserved9 | Bool | 12 | | Reserved9 | Real | 13 | |
| Reserved10 | Bool | 13 | | Reserved10 | Real | 14 | |
| Reserved11 | Bool | 14 | | Reserved11 | Real | 15 | |
| Reserved12 | Bool | 15 | | Reserved12 | Real | 16 | |

| INT16 | | | | 32 REAL | | | |
|---|------|-----|--|------------------|------|----------|--|
| Communication - Modbus (Frequency Regulation) | | | | | | | |
| Comands name | Type | bit | | Parameter Name | Type | Position | |
| Enable_DroopP | Bool | 0 | | DroopConstantP | Real | 1 | |
| Delta_Power_Limit_FreqRegulation | Bool | 1 | | Max_delta_Energy | Real | 2 | |
| Delta_Energy_Limit | Bool | 2 | | PowerLimit_Freq | Real | 3 | |
| Reset_Energy_State | Bool | 3 | | FrequencyCutOff | Real | 4 | |
| DeadBandLimitationP | Bool | 4 | | FreqDeadBand | Real | 5 | |
| Reset_To_Default_FreqRegulation | Bool | 5 | | Reserved1 | Real | 6 | |
| Reserved1 | Bool | 6 | | Reserved2 | Real | 7 | |
| Reserved2 | Bool | 7 | | Reserved3 | Real | 8 | |
| Reserved3 | Bool | 8 | | Reserved4 | Real | 9 | |
| Reserved4 | Bool | 9 | | Reserved5 | Real | 10 | |
| Reserved5 | Bool | 10 | | Reserved6 | Real | 11 | |
| Reserved6 | Bool | 11 | | Reserved7 | Real | 12 | |
| Reserved7 | Bool | 12 | | Reserved8 | Real | 13 | |
| Reserved8 | Bool | 13 | | Reserved9 | Real | 14 | |
| Reserved9 | Bool | 14 | | Reserved10 | Real | 15 | |
| Reserved10 | Bool | 15 | | Reserved11 | Real | 16 | |

| INT16 | | | 32 REAL | | |
|------------------------------------|------|-----|--------------------|------|----------|
| Communication - FLC (Common Block) | | | | | |
| Comands name | Type | bit | Parameter Name | Type | Position |
| ActivePowerLimitation | BOOL | 0 | NominalPower | REAL | 0 |
| ReactivePowerLimitation | BOOL | 1 | NominalFreq | REAL | 1 |
| Delta_Limitation | BOOL | 2 | NominaVolt | REAL | 2 |
| E_Limitation | BOOL | 3 | ActivePowerLimit | REAL | 3 |
| SOC_Limitation | BOOL | 4 | ReactivePowerLimit | REAL | 4 |
| Reset_To_Default_Common | BOOL | 5 | Delta_Limits | REAL | 5 |
| Reserved1 | BOOL | 6 | E Limits | REAL | 6 |
| Reserved2 | BOOL | 7 | Kp_ReactiveControl | REAL | 7 |
| Reserved3 | BOOL | 8 | Ki_ReactiveControl | REAL | 8 |
| Reserved4 | BOOL | 9 | VirtualReactance | REAL | 9 |
| Reserved5 | BOOL | 10 | VirtualResistance | REAL | 10 |
| Reserved6 | BOOL | 11 | Reserved1 | REAL | 11 |
| Reserved7 | BOOL | 12 | Reserved2 | REAL | 12 |
| Reserved8 | BOOL | 13 | Reserved3 | REAL | 13 |
| Reserved9 | BOOL | 14 | Reserved4 | REAL | 14 |
| Reserved10 | BOOL | 15 | Reserved5 | REAL | 15 |

| 1 (INT16) | | | 16 (32 REAL) | | | |
|---|------|-----|---------------------------|------|----------|--|
| Communication - FLC (Modal Frequency Attenuation) | | | | | | |
| Comands name | Type | bit | Parameter Name | Type | Position | |
| Enable_Modalfreq | BOOL | 0 | Gain_Modalfreq | REAL | 0 | |
| ModalFreq_PowerLimitation | BOOL | 1 | ModalFreq_PowerLimit | REAL | 1 | |
| ROCOF_Limitation_Modalfreq | BOOL | 2 | Max_ROCOF_Modalfreq | REAL | 2 | |
| Reset_To_Default_Modalfreq | BOOL | 3 | BandPassFilter_CenterFreq | REAL | 3 | |
| Reserved1 | BOOL | 4 | Reserved1 | REAL | 4 | |
| Reserved2 | BOOL | 5 | Reserved2 | REAL | 5 | |
| Reserved3 | BOOL | 6 | Reserved3 | REAL | 6 | |
| Reserved4 | BOOL | 7 | Reserved4 | REAL | 7 | |
| Reserved5 | BOOL | 8 | Reserved5 | REAL | 8 | |
| Reserved6 | BOOL | 9 | Reserved6 | REAL | 9 | |
| Reserved7 | BOOL | 10 | Reserved7 | REAL | 10 | |
| Reserved8 | BOOL | 11 | Reserved8 | REAL | 11 | |
| Reserved9 | BOOL | 12 | Reserved9 | REAL | 12 | |
| Reserved10 | BOOL | 13 | Reserved10 | REAL | 13 | |
| Reserved11 | BOOL | 14 | Reserved11 | REAL | 14 | |
| Reserved12 | BOOL | 15 | Reserved12 | REAL | 15 | |

| 1 (INT16) | | | | 16 (32 REAL) | | | |
|---|--|------|-----|---------------------|--|------|----------|
| Communication - FLC (Inertia Emulation) | | | | | | | |
| Comands name | | Type | bit | Parameter Name | | Type | Position |
| Enable_Inertia | | BOOL | 0 | Inertia_Constant | | REAL | 0 |
| Reset_States_Inertia | | BOOL | 1 | Damping_Constant | | REAL | 1 |
| ROCOF_Limitation | | BOOL | 2 | HighPassFilterConst | | REAL | 2 |
| Reset_To_Default_Inertia | | BOOL | 3 | Max_ROCOF | | REAL | 3 |
| Reserved1 | | BOOL | 4 | Reserved1 | | REAL | 4 |
| Reserved2 | | BOOL | 5 | Reserved2 | | REAL | 5 |
| Reserved3 | | BOOL | 6 | Reserved3 | | REAL | 6 |
| Reserved4 | | BOOL | 7 | Reserved4 | | REAL | 7 |
| Reserved5 | | BOOL | 8 | Reserved5 | | REAL | 8 |
| Reserved6 | | BOOL | 9 | Reserved6 | | REAL | 9 |
| Reserved7 | | BOOL | 10 | Reserved7 | | REAL | 10 |
| Reserved8 | | BOOL | 11 | Reserved8 | | REAL | 11 |
| Reserved9 | | BOOL | 12 | Reserved9 | | REAL | 12 |
| Reserved10 | | BOOL | 13 | Reserved10 | | REAL | 13 |
| Reserved11 | | BOOL | 14 | Reserved11 | | REAL | 14 |
| Reserved12 | | BOOL | 15 | Reserved12 | | REAL | 15 |

| 1 (INT16) | | | 16 (32 REAL) | | |
|--|------|-----|---------------------------|------|----------|
| Communication - FLC (Voltage Regulation) | | | | | |
| Comands name | Type | bit | Parameter Name | Type | Position |
| Enable_VoltageRegulation | BOOL | 0 | DroopConstantQ | REAL | 0 |
| Enable_DeadBandV | BOOL | 1 | LowPassFilter_CutOff_Volt | REAL | 1 |
| Delta_Power_Limitation_VoltReg | BOOL | 2 | DeadBand_Limit_Q | REAL | 2 |
| Reset_To_Default_VoltReg | BOOL | 3 | PowerLimit_Volt | REAL | 3 |
| Reserved1 | BOOL | 4 | Reserved1 | REAL | 4 |
| Reserved2 | BOOL | 5 | Reserved2 | REAL | 5 |
| Reserved3 | BOOL | 6 | Reserved3 | REAL | 6 |
| Reserved4 | BOOL | 7 | Reserved4 | REAL | 7 |
| Reserved5 | BOOL | 8 | Reserved5 | REAL | 8 |
| Reserved6 | BOOL | 9 | Reserved6 | REAL | 9 |
| Reserved7 | BOOL | 10 | Reserved7 | REAL | 10 |
| Reserved8 | BOOL | 11 | Reserved8 | REAL | 11 |
| Reserved9 | BOOL | 12 | Reserved9 | REAL | 12 |
| Reserved10 | BOOL | 13 | Reserved10 | REAL | 13 |
| Reserved11 | BOOL | 14 | Reserved11 | REAL | 14 |
| Reserved12 | BOOL | 15 | Reserved12 | REAL | 15 |

| 1 (INT16) | | | 16 (32 REAL) | | |
|--|------|-----|---------------------------|------|----------|
| Communication - FLC (Frequency Regulation) | | | | | |
| Comands name | Type | bit | Parameter Name | Type | Position |
| Enable_DroopP | BOOL | 0 | DroopConstantP | REAL | 0 |
| Enable_DeadBandFreq | BOOL | 1 | Max_Delta_Energy | REAL | 1 |
| Delta_Power_Limitation_Freq | BOOL | 2 | PowerLimit_Freq | REAL | 2 |
| Delta_Energy_Limitation | BOOL | 3 | LowPassFilter_CutOff_Freq | REAL | 3 |
| Reset_To_Default_Freq | BOOL | 4 | DeadBand_Limits_P | REAL | 4 |
| Reserved1 | BOOL | 5 | Reserved1 | REAL | 5 |
| Reserved2 | BOOL | 6 | Reserved2 | REAL | 6 |
| Reserved3 | BOOL | 7 | Reserved3 | REAL | 7 |
| Reserved4 | BOOL | 8 | Reserved4 | REAL | 8 |
| Reserved5 | BOOL | 9 | Reserved5 | REAL | 9 |
| Reserved6 | BOOL | 10 | Reserved6 | REAL | 10 |
| Reserved7 | BOOL | 11 | Reserved7 | REAL | 11 |
| Reserved8 | BOOL | 12 | Reserved8 | REAL | 12 |
| Reserved9 | BOOL | 13 | Reserved9 | REAL | 13 |
| Reserved10 | BOOL | 14 | Reserved10 | REAL | 14 |
| Reserved11 | BOOL | 15 | Reserved11 | REAL | 15 |

| INT16 | | | 32 REAL | | |
|------------------------------------|------|-----|----------------|------|----------|
| Communication - SLC (Common Block) | | | | | |
| Comands name | Type | bit | Parameter Name | Type | Position |
| Reserved1 | Bool | 0 | BattSoC | Real | 1 |
| Reserved2 | Bool | 1 | BattMaxP | Real | 2 |
| Reserved3 | Bool | 2 | BattMinP | Real | 3 |
| Reserved4 | Bool | 3 | Reserved1 | Real | 4 |
| Reserved5 | Bool | 4 | Reserved2 | Real | 5 |
| Reserved6 | Bool | 5 | Reserved3 | Real | 6 |
| Reserved7 | Bool | 6 | Reserved4 | Real | 7 |
| Reserved8 | Bool | 7 | Reserved5 | Real | 8 |
| Reserved9 | Bool | 8 | Reserved6 | Real | 9 |
| Reserved10 | Bool | 9 | Reserved7 | Real | 10 |
| Reserved11 | Bool | 10 | Reserved8 | Real | 11 |
| Reserved12 | Bool | 11 | Reserved9 | Real | 12 |
| Reserved13 | Bool | 12 | Reserved10 | Real | 13 |
| Reserved14 | Bool | 13 | Reserved11 | Real | 14 |
| Reserved15 | Bool | 14 | Reserved12 | Real | 15 |
| Reserved16 | Bool | 15 | Reserved13 | Real | 16 |

| INT16 | | | 32 REAL | | |
|--|------|-----|---------------------|------|----------|
| Communication - SLC (Resource Variability) | | | | | |
| Comands name | Type | bit | Parameter Name | Type | Position |
| Enable_RVM | Bool | 0 | KP_RVM: | Real | 1 |
| Enable_ServiceLimit | Bool | 1 | KI_RVM | Real | 2 |
| Reserved2 | Bool | 2 | MaxRate_RVM | Real | 3 |
| Reserved3 | Bool | 3 | MaxDeviation_RVM | Real | 4 |
| Reserved4 | Bool | 4 | MaxReconnection_RVM | Real | 5 |
| Reserved5 | Bool | 5 | Ts_RVM | Real | 6 |
| Reserved6 | Bool | 6 | LF_LPF_RVM | Real | 7 |
| Reserved7 | Bool | 7 | HF_HPF_RVM | Real | 8 |
| Reserved8 | Bool | 8 | RVM_Limit_Max | Real | 9 |
| Reserved9 | Bool | 9 | RVM_Limit_Min | Real | 10 |
| Reserved10 | Bool | 10 | Reserved3 | Real | 11 |
| Reserved11 | Bool | 11 | Reserved4 | Real | 12 |
| Reserved12 | Bool | 12 | Reserved5 | Real | 13 |
| Reserved13 | Bool | 13 | Reserved6 | Real | 14 |
| Reserved14 | Bool | 14 | Reserved7 | Real | 15 |
| Reserved15 | Bool | 15 | Reserved8 | Real | 16 |

| INT16 | | | 32 REAL | | |
|---|------|-----|----------------|------|----------|
| Communication - SLC (Power Shifting Management) | | | | | |
| Comands name | Type | bit | Parameter Name | Type | Position |
| Enable_PSM | Bool | 0 | KP_PSM | Real | 1 |
| Enable_ServiceLimit | Bool | 1 | KI_PSM | Real | 2 |
| Reserved2 | Bool | 2 | InputRef_PSM | Real | 3 |
| Reserved3 | Bool | 3 | Ts_PSM | Real | 4 |
| Reserved4 | Bool | 4 | PSM_Limit_Max | Real | 5 |
| Reserved5 | Bool | 5 | PSM_Limit_Min | Real | 6 |
| Reserved6 | Bool | 6 | Reserved3 | Real | 7 |
| Reserved7 | Bool | 7 | Reserved4 | Real | 8 |
| Reserved8 | Bool | 8 | Reserved5 | Real | 9 |
| Reserved9 | Bool | 9 | Reserved6 | Real | 10 |
| Reserved10 | Bool | 10 | Reserved7 | Real | 11 |
| Reserved11 | Bool | 11 | Reserved8 | Real | 12 |
| Reserved12 | Bool | 12 | Reserved9 | Real | 13 |
| Reserved13 | Bool | 13 | Reserved10 | Real | 14 |
| Reserved14 | Bool | 14 | Reserved11 | Real | 15 |
| Reserved15 | Bool | 15 | Reserved12 | Real | 16 |

| INT16 | | | 32 REAL | | |
|---|------|-----|------------------|------|------|
| Communication - SLC (Ramp Rate Enhancement) | | | | | |
| Command | Type | Bit | Parameter | Type | Data |
| Enable_RRE | Bool | 0 | KP_RRE | Real | 1 |
| Enable_ServiceLimit | Bool | 1 | KI_RRE | Real | 2 |
| Reserved2 | Bool | 2 | MaxRate_RRE | Real | 3 |
| Reserved3 | Bool | 3 | RateLimited_RRE | Real | 4 |
| Reserved4 | Bool | 4 | Ts_RRE | Real | 5 |
| Reserved5 | Bool | 5 | SampligFrequency | Real | 6 |
| Reserved6 | Bool | 6 | RRE_Limit_Max | Real | 7 |
| Reserved7 | Bool | 7 | RRE_Limit_Min | Real | 8 |
| Reserved8 | Bool | 8 | Reserved3 | Real | 9 |
| Reserved9 | Bool | 9 | Reserved4 | Real | 10 |
| Reserved10 | Bool | 10 | Reserved5 | Real | 11 |
| Reserved11 | Bool | 11 | Reserved6 | Real | 12 |
| Reserved12 | Bool | 12 | Reserved7 | Real | 13 |
| Reserved13 | Bool | 13 | Reserved8 | Real | 14 |
| Reserved14 | Bool | 14 | Reserved9 | Real | 15 |
| Reserved15 | Bool | 15 | Reserved10 | Real | 16 |

| INT16 | | | 32 REAL | | |
|---|------|-----|-------------------|------|------|
| Communication - SLC (Power Curtailment) | | | | | |
| Command | Type | Bit | Parameter | Type | Data |
| Enable_PCM | Bool | 0 | KP_PCM | Real | 1 |
| Enable_PCM_SP | Bool | 1 | KI_PCM | Real | 2 |
| Enable_PCM_APL | Bool | 2 | BalanceControlRef | Real | 3 |
| Enable_PCM_DC | Bool | 3 | DeltaControlRef | Real | 4 |
| Enable_PCM_SR | Bool | 4 | StopRegulationP | Real | 5 |
| Enable_ServiceLimit | Bool | 5 | Ts_PCM | Real | 6 |
| Reserved2 | Bool | 6 | PCM_Limit_Max | Real | 7 |
| Reserved3 | Bool | 7 | PCM_Limit_Min | Real | 8 |
| Reserved4 | Bool | 8 | Reserved3 | Real | 9 |
| Reserved5 | Bool | 9 | Reserved4 | Real | 10 |
| Reserved6 | Bool | 10 | Reserved5 | Real | 11 |
| Reserved7 | Bool | 11 | Reserved6 | Real | 12 |
| Reserved8 | Bool | 12 | Reserved7 | Real | 13 |
| Reserved9 | Bool | 13 | Reserved8 | Real | 14 |
| Reserved10 | Bool | 14 | Reserved9 | Real | 15 |
| Reserved11 | Bool | 15 | Reserved10 | Real | 16 |

| INT16 | | | 32 REAL | | |
|-------------------------------------|------|-----|--------------------|------|------|
| Communication - SLC (Power Balance) | | | | | |
| Command | Type | Bit | Parameter | Type | Data |
| Enable_PBP | Bool | 0 | KP_PBP | Real | 1 |
| Enable_ServiceLimit | Bool | 1 | KI_PBP | Real | 2 |
| Reserved2 | Bool | 2 | LinePowerReference | Real | 3 |
| Reserved3 | Bool | 3 | FrequencyReference | Real | 4 |
| Reserved4 | Bool | 4 | Ts_PBP | Real | 5 |
| Reserved5 | Bool | 5 | PBP Limit_Max | Real | 6 |
| Reserved6 | Bool | 6 | PBP Limit_Min | Real | 7 |
| Reserved7 | Bool | 7 | Reserved3 | Real | 8 |
| Reserved8 | Bool | 8 | Reserved4 | Real | 9 |
| Reserved9 | Bool | 9 | Reserved5 | Real | 10 |
| Reserved10 | Bool | 10 | Reserved6 | Real | 11 |
| Reserved11 | Bool | 11 | Reserved7 | Real | 12 |
| Reserved12 | Bool | 12 | Reserved8 | Real | 13 |
| Reserved13 | Bool | 13 | Reserved9 | Real | 14 |
| Reserved14 | Bool | 14 | Reserved10 | Real | 15 |
| Reserved15 | Bool | 15 | Reserved11 | Real | 16 |